

# A Unifying Framework for Sparse Gaussian Process Approximation using Power Expectation Propagation

Thang D. Bui

TDB40@CAM.AC.UK

Josiah Yan

JOSIAH.YAN@GMAIL.COM

Richard E. Turner

RET26@CAM.AC.UK

*Computational and Biological Learning Lab, Department of Engineering  
University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK*

## Abstract

Gaussian processes (GPs) are flexible distributions over functions that enable high-level assumptions about unknown functions to be encoded in a parsimonious, flexible and general way. Although elegant, the application of GPs is limited by computational and analytical intractabilities that arise when data are sufficiently numerous or when employing non-Gaussian models. Consequently, a wealth of GP approximation schemes have been developed over the last 15 years to address these key limitations. Many of these schemes employ a small set of pseudo data points to summarise the actual data. In this paper we develop a new pseudo-point approximation framework using Power Expectation Propagation (Power EP) that unifies a large number of these pseudo-point approximations. Unlike much of the previous venerable work in this area, the new framework is built on standard methods for approximate inference (variational free-energy, EP and power EP methods) rather than employing approximations to the probabilistic generative model itself. In this way all of approximation is performed at ‘inference time’ rather than at ‘modelling time’ resolving awkward philosophical and empirical questions that trouble previous approaches. Crucially, we demonstrate that the new framework includes new pseudo-point approximation methods that outperform current approaches on regression, classification and state space modelling tasks.

## 1. Introduction

Gaussian Processes (GPs) are powerful nonparametric distributions over continuous functions that are routinely deployed in probabilistic modelling for applications ranging from regression and classification (Rasmussen and Williams, 2005), representation learning (Lawrence, 2005), state space modelling (Wang et al., 2005), active learning (Houlsby et al., 2011), reinforcement learning (Deisenroth, 2010), black-box optimisation (Snoek et al., 2012), and numerical methods (Mahsereci and Hennig, 2015). GPs have many elegant theoretical properties, but their use is greatly hindered by analytic and computational intractabilities. A large research effort has been directed at this fundamental problem resulting in the development of a plethora of sparse approximation methods that can sidestep these intractabilities (Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006; Snelson, 2007; Schwaighofer and Tresp, 2002; Titsias, 2009b; Csató, 2002; Csató and Opper, 2002; Seeger et al., 2003; Naish-Guzman and Holden, 2007; Qi et al., 2010; Bui and Turner, 2014; Hens-

man et al., 2015; Hernández-Lobato and Hernández-Lobato, 2016; Matthews et al., 2016; Figueiras-Vidal and Lázaro-Gredilla, 2009; Frigola et al., 2014; McHutchon, 2014)

This paper develops a general sparse approximate inference framework based upon Power Expectation Propagation (PEP) (Minka, 2004) that unifies many of these approximations, extends them significantly, and provides improvements in practical settings. In this way, the paper provides a complementary perspective to the seminal review of Quiñero-Candela and Rasmussen (2005) viewing sparse approximations through the lens of approximate *inference*, rather than approximate *generative models*.

The paper begins by reviewing several frameworks for sparse approximation focussing on the GP regression and classification setting (section 2). It then lays out the new unifying framework and the relationship to existing techniques (section 3). The extension to state space models follows (section 4). A thorough experimental evaluation is presented in section 5.

## 2. Pseudo-point Approximations for GP Regression and Classification

This section provides a concise introduction to GP regression and classification and then reviews several pseudo-point based sparse approximation schemes for these models. For simplicity, we first consider a supervised learning setting in which the training set comprises  $N$   $D$ -dimensional input and scalar output pairs  $\{\mathbf{x}_n, y_n\}_{n=1}^N$  and the goal is to produce probabilistic predictions for the outputs corresponding to novel inputs. A non-linear function,  $f(\mathbf{x})$ , can be used to parameterise the probabilistic mapping between inputs and outputs,  $p(y_n|f, \mathbf{x}_n, \theta)$ . Typical choices for the probabilistic mapping are Gaussian  $p(y_n|f, \mathbf{x}_n, \theta) = \mathcal{N}(y_n; f(\mathbf{x}_n), \sigma_y^2)$  for the regression setting ( $y_n \in \mathbb{R}$ ) and Bernoulli  $p(y_n|f, \mathbf{x}_n, \theta) = \mathcal{B}(y_n; \Phi(f(\mathbf{x}_n)))$  with a sigmoidal link function  $\Phi(f)$  for the binary classification setting ( $y_n \in \{0, 1\}$ ). Whilst it is possible to specify the non-linear function  $f$  via an explicit parametric form, a more flexible and elegant approach employs a GP prior over the functions directly,  $p(f|\theta) = \mathcal{GP}(f; 0, k_\theta(\cdot, \cdot))$ , here assumed without loss of generality to have a zero mean-function and a covariance function  $k_\theta(\mathbf{x}, \mathbf{x}')$ . This class of probabilistic models has a joint distribution

$$p(f, \mathbf{y}|\theta) = p(f|\theta) \prod_{n=1}^N p(y_n|f(\mathbf{x}_n), \theta) \quad (1)$$

where we have collected the observations into the vector  $\mathbf{y}$  and suppressed the inputs on the left hand side to lighten the notation.

This model class contains two potential sources of intractability. First, the possibly non-linear likelihood function can introduce analytic intractabilities that require approximation. Second, the GP prior entails an  $\mathcal{O}(N^3)$  complexity that is computationally intractable for many practical problems. These two types of intractability can be handled by combining standard approximate inference methods with pseudo-point approximations that summarise the full Gaussian process via  $M$  pseudo data points leading to an  $\mathcal{O}(NM^2)$  cost. The main approaches of this sort can be characterised in terms of two parallel frameworks that are described in the following sections.

## 2.1 Sparse GP Approximation via Approximate Generative Models

The first framework begins by constructing a new generative model that is similar to the original, so that inference in the new model might be expected to produce similar results, but which has a special structure that supports efficient computation. Typically this approach involves approximating the Gaussian process prior as it is the origin of the cubic cost (Quiñonero-Candela and Rasmussen, 2005).

The seminal review by Quiñonero-Candela and Rasmussen (Quiñonero-Candela and Rasmussen, 2005) reinterprets a family of approximations in terms of this unifying framework. The GP prior is approximated by identifying a small set of  $M \leq N$  pseudo-points  $\mathbf{u}$ , here assumed to be disjoint from the training function values  $\mathbf{f}$  so that  $f = \{\mathbf{u}, \mathbf{f}, f_{\neq \mathbf{u}, \mathbf{f}}\}$ . The GP prior is then decomposed using the product rule

$$p(f|\theta) = p(\mathbf{u}|\theta)p(\mathbf{f}|\mathbf{u}, \theta)p(f_{\neq \mathbf{u}, \mathbf{f}}|\mathbf{f}, \mathbf{u}, \theta). \quad (2)$$

Of central interest is the relationship between the pseudo-points and the training function values  $p(\mathbf{f}|\mathbf{u}, \theta) = \mathcal{N}(\mathbf{f}; \mathbf{K}_{\mathbf{f}\mathbf{u}}\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{u}, \mathbf{D}_{\mathbf{f}\mathbf{f}})$  where  $\mathbf{D}_{\mathbf{f}\mathbf{f}} = \mathbf{K}_{\mathbf{f}\mathbf{f}} - \mathbf{Q}_{\mathbf{f}\mathbf{f}}$  and  $\mathbf{Q}_{\mathbf{f}\mathbf{f}} = \mathbf{K}_{\mathbf{f}\mathbf{u}}\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u}\mathbf{f}}$ . Here we have introduced matrices corresponding to the covariance function's evaluation at the pseudo-input locations  $\{\mathbf{z}_m\}_{m=1}^M$ , so that  $[\mathbf{K}_{\mathbf{u}\mathbf{u}}]_{mm'} = k_\theta(\mathbf{z}_m, \mathbf{z}_{m'})$  and similarly for the covariance between the pseudo-input and data locations  $[\mathbf{K}_{\mathbf{u}\mathbf{f}}]_{mn} = k_\theta(\mathbf{z}_m, \mathbf{x}_n)$ . Importantly, this term saddles learning with a cubic complexity cost. Computationally efficient approximations can be constructed by simplifying these dependencies between the pseudo-points and the data function values  $q(\mathbf{f}|\mathbf{u}, \theta) \approx p(\mathbf{f}|\mathbf{u}, \theta)$ . In order to benefit from these efficiencies at prediction time as well, a second approximation is made whereby the pseudo-points form a bottleneck between the data function values and test function values  $p(f_{\neq \mathbf{u}, \mathbf{f}}|\mathbf{u}, \theta) \approx p(f_{\neq \mathbf{u}, \mathbf{f}}|\mathbf{f}, \mathbf{u}, \theta)$ . Together, the two approximations result in an approximate prior process,

$$q(f|\theta) = p(\mathbf{u}|\theta)q(\mathbf{f}|\mathbf{u}, \theta)p(f_{\neq \mathbf{u}, \mathbf{f}}|\mathbf{f}, \mathbf{u}, \theta). \quad (3)$$

We can now compactly summarise a number of previous approaches to GP approximation as special cases of the choice

$$q(\mathbf{f}|\mathbf{u}, \theta) = \prod_{b=1}^B \mathcal{N}(\mathbf{f}_b; \mathbf{K}_{\mathbf{f}_b, \mathbf{u}}\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{u}, \alpha \mathbf{D}_{\mathbf{f}_b, \mathbf{f}_b}) \quad (4)$$

where  $b$  indexes  $B$  disjoint blocks of data-function values. The Deterministic Training Conditional (DTC) approximation uses  $\alpha \rightarrow 0$ ; the Fully Independent Training Conditional (FITC) approximation uses  $\alpha = 1$  and  $B = N$ ; the Partially Independent Training Conditional (PITC) approximation uses  $\alpha = 1$  (Quiñonero-Candela and Rasmussen, 2005; Schwaighofer and Tresp, 2002).

In a moment we will consider inference in the modified models, before doing so we note that it is possible to construct more flexible modified prior processes using the inter-domain approach that places the pseudo-points in a different domain from the data, defined by a linear integral transform  $g(z) = \int w(z, z')f(z')dz'$ . Here the window  $w(z, z')$  might be a Gaussian blur or a wavelet transform. The pseudo-points are now placed in the new domain  $g = \{\mathbf{u}, \mathbf{g}_{\neq \mathbf{u}}\}$  where they induce a potentially more flexible Gaussian process in the

old domain  $f$  through the linear transform (see Figueiras-Vidal and Lázaro-Gredilla (2009) for FITC). The expressions in this section still hold, but the covariance matrices involving pseudo-points are modified to take account of the transform,

$$[\mathbf{K}_{\mathbf{uu}}]_{mm'} = \int w(\mathbf{z}_m, \mathbf{z}) k_\theta(\mathbf{z}, \mathbf{z}') w(\mathbf{z}', \mathbf{z}_{m'}) d\mathbf{z} d\mathbf{z}', \quad [\mathbf{K}_{\mathbf{uf}}]_{mn} = \int w(\mathbf{z}_m, \mathbf{z}) k_\theta(\mathbf{z}, \mathbf{x}_n) d\mathbf{z}. \quad (5)$$

Having specified modified prior processes, these can be combined with the original likelihood function to produce a new generative models. In the case of point-wise likelihoods we have

$$q(\mathbf{y}, f|\theta) = q(f|\theta) \prod_{n=1}^N p(y_n|f(\mathbf{x}_n), \theta). \quad (6)$$

Inference and learning can now be performed using the modified model using standard techniques. Due to the form of the new prior process, the computational complexity is  $\mathcal{O}(NM^2)$  (for testing,  $N$  becomes the number of test data points, assuming dependencies between the test-points are not computed).<sup>1</sup> For example, in the case of regression, the posterior distribution over function values  $f$  (necessary for inference and prediction) has a simple analytic form

$$q(f|\mathbf{y}, \theta) = \mathcal{GP}(f; \mu_{f|\mathbf{y}}, \Sigma_{f|\mathbf{y}}), \quad \mu_{f|\mathbf{y}} = \mathbf{Q}_{ff} \bar{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{y}, \quad \Sigma_{f|\mathbf{y}} = \mathbf{K}_{ff} - \mathbf{Q}_{ff} \bar{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{Q}_{ff} \quad (7)$$

where  $\bar{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \text{blkdiag}(\{\alpha_b \mathbf{D}_{\mathbf{f}_b \mathbf{f}_b}\}_{b=1}^B) + \sigma_y^2 \mathbf{I}$  and  $\text{blkdiag}$  builds a block-diagonal matrix from its inputs. One way of understanding the origin of the computational gains is that the new generative model corresponds to a form of factor analysis in which the  $M$  pseudo-points determine the  $N$  function values at the observed data (as well as at potential test locations) via a linear Gaussian relationship. This results in low rank (sparse) structure in  $\bar{\mathbf{K}}_{\mathbf{ff}}$  that can be exploited through the matrix inversion and determinant lemmas. In the case of regression, the new model's marginal likelihood also has an analytic form that allows the hyper-parameters,  $\theta$ , to be learned via optimisation

$$\log q(\mathbf{y}|\theta) = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\bar{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2} \mathbf{y}^\top \bar{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{y}. \quad (8)$$

The approximate generative model framework has attractive properties. The cost of inference, learning, and prediction has been reduced from  $\mathcal{O}(N^3)$  to  $\mathcal{O}(NM^2)$  and in many cases accuracy can be maintained with a relatively small number of pseudo-points. The pseudo-point input locations can be optimised by maximising the new model's marginal likelihood (Snelson and Ghahramani, 2006). When  $M = N$  and the pseudo-points and observed data inputs coincide, then FITC and PITC are exact which appears reassuring. However, the framework is philosophically challenging as the elegant separation of model and (approximate) inference has been lost. Are we allowed, for example, to add new pseudo-points as more data are acquired and the complexity of the underlying function is revealed? This seems sensible, but effectively changes the modelling assumptions as more data are

---

1. It is assumed that the maximum size of the blocks is not greater than the number of pseudo-points  $\dim(\mathbf{f}_b) \leq M$ .

seen. Similarly, if the pseudo-input locations are optimised, the principled non-parametric model has suddenly acquired  $MD$  parameters and with them all of the concomitant issues of parametric models including overfitting and optimisation difficulties (Bauer et al., 2016). Finally, for analytically intractable likelihood functions an additional approximate inference step is required anyway, begging the question; why not handle computational and analytic intractabilities together at inference time?

## 2.2 Sparse GP Approximation via Approximate Inference: VFE

The approximate generative model framework for constructing sparse approximations is philosophically troubling. In addition, learning pseudo-point input locations via optimisation of the model likelihood can perform poorly e.g. for DTC it is prone to overfitting even for  $M \ll N$  (Titsias, 2009b). This motivates a more direct approach that commits to the true generative model and performs all of the necessary approximation at inference time.

Perhaps the most well known approach in this vein is Titsias’s beautiful sparse variational free energy (VFE) method (Titsias, 2009b). The original presentation of this work employs finite variable sets and an augmentation trick that arguably obscures its full elegance. Here instead we follow Matthews et al. (2016) and lower bound the marginal likelihood using a distribution  $q(f)$  over the entire infinite dimensional function,

$$\log p(\mathbf{y}|\theta) = \log \int p(\mathbf{y}, f|\theta) df \geq \int q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} df = \mathbb{E}_{q(f)} \left[ \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} \right] = \mathcal{F}(q, \theta).$$

The VFE bound can be written as the difference between the model log-marginal likelihood and the KL divergence between the variational distribution and the true posterior  $\mathcal{F}(q, \theta) = \log p(\mathbf{y}|\theta) - \text{KL}(q(f)||p(f|\mathbf{y}, \theta))$ . The bound is therefore saturated when  $q(f) = p(f|\mathbf{y}, \theta)$ , but this is intractable. Instead, pseudo-points are made explicit,  $f = \{\mathbf{u}, f_{\neq \mathbf{u}}\}$ , and an approximate posterior distribution used of the following form  $q(f) = q(\mathbf{u}, f_{\neq \mathbf{u}}|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$ . Under this approximation, the set of variables  $f_{\neq \mathbf{u}}$  do not experience the data directly, but rather only through the pseudo-points, as can be seen by comparison to the true posterior  $p(f|\mathbf{y}, \theta) = p(f_{\neq \mathbf{u}}|\mathbf{y}, \mathbf{u}, \theta)p(\mathbf{u}|\mathbf{y}, \theta)$ . Importantly, the form of the approximate posterior causes a cancellation of the prior conditional term, which gives rise to a bound with  $\mathcal{O}(NM^2)$  complexity,

$$\begin{aligned} \mathcal{F}(q, \theta) &= \mathbb{E}_{q(f|\theta)} \left[ \frac{\log p(\mathbf{y}|f, \theta) \cancel{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)} p(\mathbf{u}|\theta)}{\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)} q(\mathbf{u})} \right] \\ &= \sum_n \mathbb{E}_{q(f|\theta)} [\log p(y_n|f_n, \theta)] - \text{KL}(q(\mathbf{u})||p(\mathbf{u}|\theta)). \end{aligned}$$

For regression, the calculus of variations can be used to find the optimal approximate posterior Gaussian process over pseudo-data  $q^{\text{opt}}(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q^{\text{opt}}(\mathbf{u})$  which has the form

$$q^{\text{opt}}(f|\theta) = \mathcal{GP}(f; \mu_{f|\mathbf{y}}, \Sigma_{f|\mathbf{y}}), \quad \mu_{f|\mathbf{y}} = \mathbf{Q}_{f\mathbf{f}} \tilde{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{y}, \quad \Sigma_{f|\mathbf{y}} = \mathbf{K}_{ff} - \mathbf{Q}_{f\mathbf{f}} \tilde{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{Q}_{\mathbf{f}f} \quad (9)$$

where  $\tilde{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \sigma_y^2 \mathbf{I}$ . This process is identical to that recovered when performing exact inference under the DTC approximate regression generative model (Titsias, 2009b) (see

equation 7). In fact DTC was originally derived using a related KL argument (Csató, 2002; Seeger et al., 2003). The optimised free-energy is

$$\mathcal{F}(q^{\text{opt}}, \theta) = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\tilde{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2} \mathbf{y}^\top \tilde{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{y} - \frac{1}{2\sigma_y^2} \text{trace}(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}). \quad (10)$$

Notice that the free-energy has an additional trace term as compared to the marginal likelihood obtained from the DTC generative model approach (see equation 8 as  $\alpha \rightarrow 0$ ). The trace term is proportional to the sum of the variances of the training function values given the pseudo-points,  $p(\mathbf{f}|\mathbf{u})$ , it thereby encourages pseudo-input locations that explain the observed data well. This term acts as a regulariser that prevents overfitting which plagues the generative model formulation of DTC.

The VFE approach can be extended to non-linear models including classification (Hensman et al., 2015), latent variable models (Titsias and Lawrence, 2010) and state space models (Frigola et al., 2014; McHutchon, 2014) by restricting  $q(\mathbf{u})$  to be Gaussian and optimising its parameters. Indeed, this uncollapsed form of the bound can be beneficial in the context of regression too as it is amenable to stochastic optimisation (Hensman et al., 2013). Additional approximation is sometimes required to compute any remaining intractable non-linear integrals, but these are often low-dimensional. For example, when the likelihood depends on only one latent function value, as is typically the case for regression and classification, the bound requires only 1D integrals  $\mathbb{E}_{q(f_n)} [\log p(y_n | f_n, \theta)]$  that can be evaluated using quadrature (Hensman et al., 2015), for example.

The VFE approach can also be extended to employ inter-domain variables (Tobar et al., 2015; Matthews et al., 2016). The approach considers the augmented generative model  $p(f, g | \theta)$  where to remind the reader the auxiliary process is defined by a linear integral transformation,  $g(z) = \int w(z, z') f(z') dz'$ . Variational inference is now performed over both latent processes  $q(f, g) = q(f, \mathbf{u}, g_{\neq \mathbf{u}} | \theta) = p(f, g_{\neq \mathbf{u}} | \mathbf{u}, \theta) q(\mathbf{u})$ . Here the pseudo-data have been placed into the auxiliary process with the idea being that they can induce richer dependencies in the original domain that model the true posterior more accurately. In fact, if the linear integral transformation is parameterised then the transformation can be learned so that it approximates the posterior more accurately.

A key concept underpinning the VFE framework is that the pseudo-input locations are purely parameters of the approximate posterior (‘variational parameters’). Optimisation of these parameters is automatically protected from overfitting since it is equivalent to minimising the KL divergence between the approximate and true posterior. Indeed, although the DTC posterior is recovered in the regression setting, as we have seen the free-energy is *not* equal to the log-marginal likelihood of the DTC generative model, containing an additional term that substantially improves the quality of the optimised pseudo-point input locations. The fact that the form of the DTC approximation can be recovered from a direct approximate inference approach and that this new perspective leads to superior pseudo-input optimisation, raises the question; can this also be done for FITC and PITC?

### 2.3 Sparse GP Approximation via Approximate Inference: EP

Expectation Propagation (EP) is a deterministic inference method (Minka, 2001) that is known to outperform VFE methods in GP classification when unsparisified fully-factored approximations  $q(\mathbf{f}) = \prod_n q_n(f_n)$  are used (Nickisch and Rasmussen, 2008). Motivated by this

observation, EP has been combined with the approximate generative modelling approach to handle non-linear likelihoods (Naish-Guzman and Holden, 2007; Hernández-Lobato and Hernández-Lobato, 2016). This begs the question: can the sparsification and the non-linear approximation be handled in a single EP inference stage, as for VFE? Astonishingly Csató and Opper not only developed such a method in 2002 (Csató and Opper, 2002), predating much of the work mentioned above, they showed that it is equivalent to applying the FITC approximation and running EP if further approximation is required. In our view, this is a central result, but it appears to have been largely overlooked by the field. Snelson was made aware of it when writing his thesis (Snelson, 2007), briefly acknowledging Csató and Opper’s contribution. Qi et al. (2010) extended Csató and Opper’s work to utilise inter-domain pseudo-points and they additionally recognised that the EP energy function at convergence is equal to the FITC log-marginal likelihood approximation. Interesting, no additional term arises as it does when the VFE approach generalised the DTC generative model approach. We are unaware of other work in this vein.

It is hard to be known for certain why these important results are not widely known, but a contributing factor is that the exposition in these papers is largely at Marr’s algorithmic level (Dawson, 1998), and does not focus on the computational level making them challenging to understand. Moreover, Csató and Opper’s paper was written before EP was formulated in a general way and the presentation, therefore, does not follow what has become the standard approach. In fact, as the focus was online inference, Assumed Density Filtering was employed rather than full-blown EP. One of the main contributions of this paper is to provide a clear computational exposition including an explicit form of the approximating distribution and full details about each step of the EP procedure. In addition, to bringing clarity we make the following novel contributions:

- We show that a generalisation of EP called Power EP can subsume the EP and VFE approaches (and therefore FITC and DTC) into a single unified framework. More precisely, the fixed points of Power EP yield the FITC and VFE posterior distribution under different limits and the Power EP marginal likelihood estimate (the negative ‘Power EP energy’) recover the FITC marginal likelihood and the VFE too. Critically the connection to the VFE method leans on the new interpretation of Titsias’s approach (Matthews et al., 2016) outlined in the previous section that directly employs the approximate posterior over function values (rather than augmenting the model with pseudo-points). The connection therefore also requires a formulation power EP that involves KL divergence minimisation between stochastic processes.
- We show how versions of PEP that are intermediate between the existing VFE and EP approaches can be derived, as well as mixed approaches that treat some data variationally and others using EP. We also show how PITC emerges from the same framework and how to incorporate inter-domain transforms. For regression, we obtain analytical expressions for the fixed points of Power EP in a general case that includes all of these extensions as well as the form of the Power EP marginal likelihood estimate at convergence that is useful for hyper-parameter and pseudo-input optimisation.
- We consider (Gaussian) regression and probit classification as canonical models on which to test the new framework and demonstrate through exhaustive testing that

versions of PEP intermediate between VFE and EP perform substantially better on average. The experiments also shed light on situations where VFE is to be preferred to EP and vice versa which is an important open area of research. We end by extending the theoretical framework and the experimental comparisons to GP state-space-models where there is less prior work.

Many of the new theoretical contributions described above are summarised in figure 1 along with their relationship to previous work.

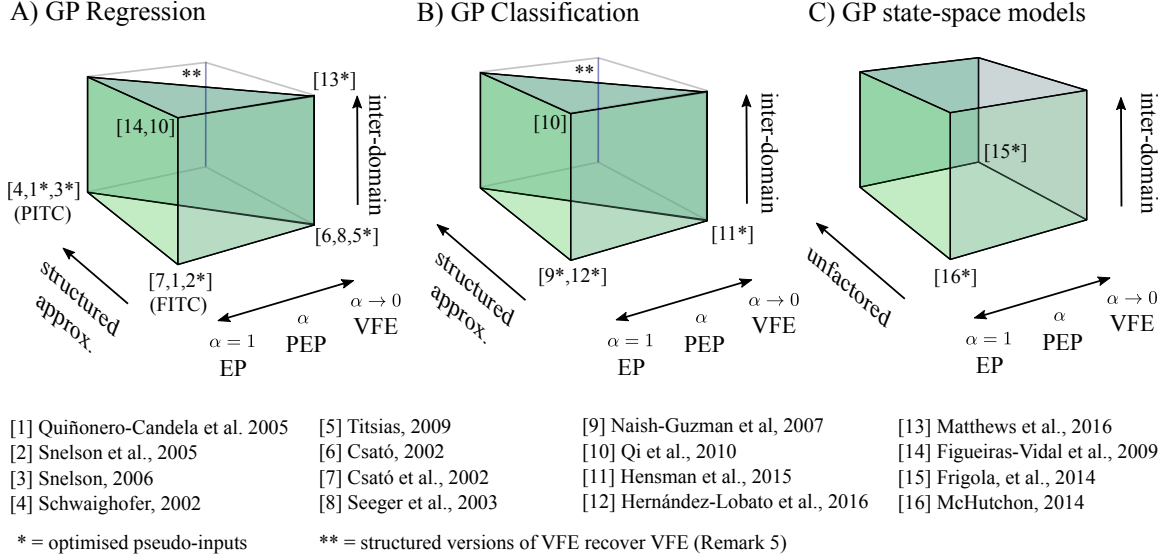


Figure 1: A unified view of pseudo-point GP approximations applied to A) regression, B) classification, and C) state-space modelling. Every point in the algorithm polygons corresponds to a form of GP approximation. Previous algorithms correspond to labelled vertices. The new Power EP framework encompasses the three polygons, including their interior.

### 3. A New Unifying View using Power Expectation Propagation

In this section, we provide a new unifying view of sparse approximation using Power Expectation Propagation (PEP or Power EP) (Minka, 2004). We review Power EP, describe how to apply it for sparse GP regression and classification, and then discuss its relationship to existing methods.

#### 3.1 Power EP

One way of understanding the goal of distributional inference approximations, including the VFE method, EP and Power EP, is that they return an approximation of a tractable form to the model *joint-distribution* evaluated on the observed data. In the case of GP regression and classification, this means  $q^*(f|\theta) \approx p(f, \mathbf{y}|\theta)$  where \* is used to denote an unnor-



malised process. Why is the model joint-distribution a sensible object of approximation? The joint distribution can be decomposed into the product of the posterior distribution and the marginal likelihood,  $p(f, \mathbf{y}|\theta) = p^*(f|\mathbf{y}, \theta) = p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)$ , the two inferential objects of interest. A tractable approximation to the joint can therefore be similarly decomposed  $q^*(f|\theta) = Zq(f|\theta)$  into a normalised component that approximates the posterior  $q(f|\theta) \approx p(f|\mathbf{y}, \theta)$  and the normalisation constant which approximates the marginal likelihood  $Z \approx p(\mathbf{y}|\theta)$ . In other words, the approximation of the joint simultaneously returns approximations to the posterior and marginal likelihood. In the current context tractability of the approximating family means that it is analytically integrable and that this integration can be performed with an appropriate computational complexity. We consider the approximating family comprising unnormalised GPs,  $q^*(f|\theta) = Z\mathcal{GP}(m_{\mathbf{f}}, V_{\mathbf{ff}'})$ .

The VFE approach can be reformulated in the new context using the un-normalised KL divergence (Zhu and Rohwer, 1997) to measure the similarity between the approximation and the joint distribution

$$\overline{\text{KL}}(q^*(f|\theta)||p(f, \mathbf{y}|\theta)) = \int q^*(f) \log \frac{q^*(f)}{p(f, \mathbf{y}|\theta)} df + \int (p(f, \mathbf{y}|\theta) - q^*(f)) df. \quad (11)$$

The un-normalised KL divergence generalises the KL divergence to accommodate un-normalised densities. It is always non-negative and collapses back to the standard form when its arguments are normalised. Minimising the un-normalised KL with respect to  $q^*(f|\theta) = Z_{\text{VFE}}q(f)$  encourages the approximation to match both the posterior and marginal-likelihood, and it yields analytic solutions

$$q^{\text{opt}}(f) = \underset{q(f) \in \mathcal{Q}}{\text{argmin}} \text{KL}(q(f)||p(f|\mathbf{y}, \theta)), \text{ and } Z_{\text{VFE}}^{\text{opt}} = \exp(\mathcal{F}(q^{\text{opt}}(f), \theta)). \quad (12)$$

That is, the standard variational free-energy approximation to the posterior and marginal likelihood is recovered. One of the pedagogical advantages of framing VFE in this way is that approximation of the posterior and marginal likelihood are committed to upfront, in contrast to the traditional derivation which begins by targeting approximation of the marginal likelihood, but shows that approximation of the posterior emerges as an essential part of this scheme (see section 2.2).

Power EP also approximates the joint-distribution employing an approximating family whose form mirrors that of the target,

$$p^*(f|\mathbf{y}, \theta) = p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta) = p(f|\theta) \prod_n p(y_n|f, \theta) \approx p(f|\theta) \prod_n t_n(\mathbf{u}) = q^*(f|\theta). \quad (13)$$

Here, the approximation retains the exact prior, but each likelihood term in the exact posterior,  $p(y_n|\mathbf{f}_n, \theta)$ , is approximated by a simple factor  $t_n(\mathbf{u})$  that is assumed Gaussian. These simple factors will be iteratively refined by the PEP algorithm such that they will capture the effect that each true likelihood has on the posterior.

Before describing the details of the PEP algorithm, it is illuminating to consider an alternative interpretation of the approximation. Together, the approximate likelihood functions specify an un-normalised Gaussian over the pseudo-points that can be written  $\prod_n t_n(\mathbf{u}) = \mathcal{N}(\tilde{\mathbf{y}}; \tilde{\mathbf{W}}\mathbf{u}, \tilde{\Sigma})$ . The approximate posterior above can therefore be thought of as the (exact) GP posterior resulting from a surrogate regression problem with surrogate

observations  $\tilde{\mathbf{y}}$  that are generated from linear combinations of the pseudo-points and additive surrogate noise  $\tilde{\mathbf{y}} = \tilde{\mathbf{W}}\mathbf{u} + \tilde{\Sigma}^{1/2}\epsilon$ . The PEP algorithm will iteratively refine  $\{\tilde{\mathbf{y}}, \tilde{\mathbf{W}}, \tilde{\Sigma}\}$  such that exact inference in the simple surrogate regression model returns a posterior and marginal likelihood estimate that is ‘close’ to that returned by performing exact inference in the intractable complex model (see figure 2).

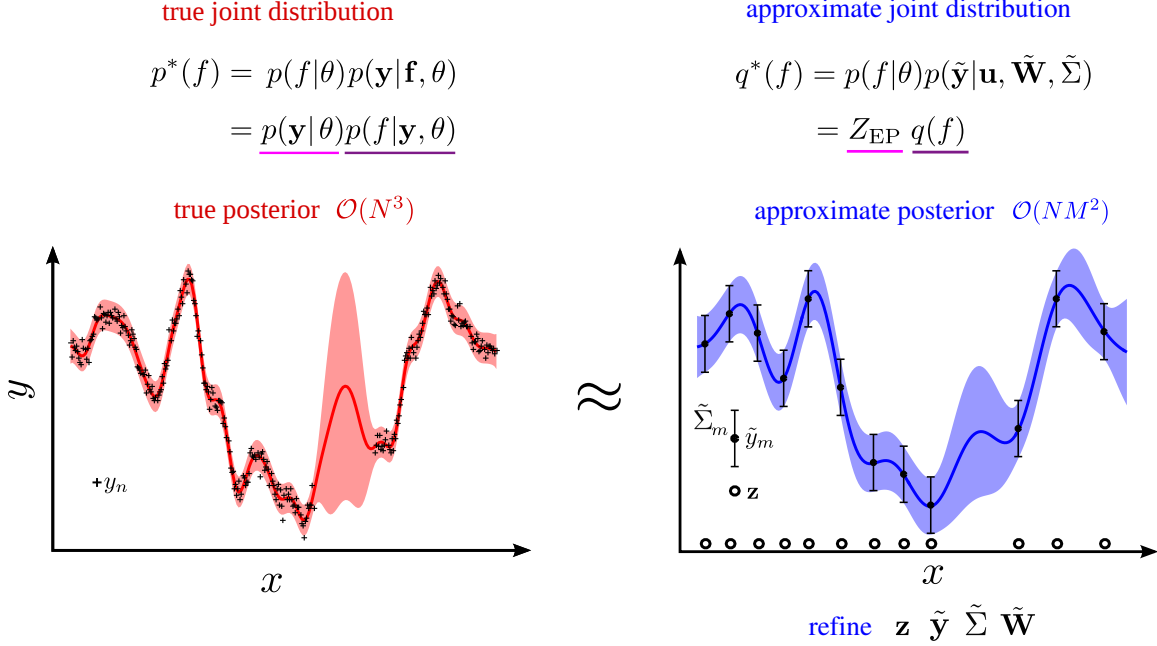


Figure 2: Perspectives on the approximating family. The true joint distribution over the unknown function  $f$  and the  $N$  data points  $\mathbf{y}$  (top left) comprises the GP prior and an intractable likelihood function. This is approximated by a surrogate regression model with a joint distribution over the function  $f$  and  $M$  surrogate data points  $\tilde{\mathbf{y}}$  (top right). The surrogate regression model employs the same GP prior, but uses a Gaussian likelihood function  $p(\tilde{\mathbf{y}}|\mathbf{u}, \tilde{\mathbf{W}}, \tilde{\Sigma}) = \mathcal{N}(\tilde{\mathbf{y}}; \tilde{\mathbf{W}}\mathbf{u}, \tilde{\Sigma})$ . The intractable true posterior (bottom left) is approximated by refining the surrogate data  $\tilde{\mathbf{y}}$  their input locations  $\mathbf{z}$  and the parameters of the surrogate model  $\tilde{\mathbf{W}}$  and  $\tilde{\Sigma}$ .

Power EP iteratively refines the approximate factors or surrogate likelihoods so that the GP posterior of the surrogate regression task best approximates the posterior of the original regression/classification problem using the following steps:

1. **Deletion:** compute the cavity distribution by removing a fraction of one approximate factor,  $q^{\setminus n}(f|\theta) \propto q^*(f|\theta)/t_n^\alpha(\mathbf{u})$ .
2. **Projection:** first, compute the tilted distribution by incorporating a corresponding fraction of the true likelihood into the cavity,  $\tilde{p}(f) = q^{\setminus n}(f|\theta)p^\alpha(y_n|\mathbf{f}_n)$ . Second, project the tilted distribution onto the approximate posterior using the KL divergence for unnormalised densities,

$$q^*(f|\theta) \leftarrow \underset{q^*(f|\theta) \in \mathcal{Q}}{\operatorname{argmin}} \overline{\text{KL}}(\tilde{p}(f)||q^*(f|\theta)). \quad (14)$$

Here  $\mathcal{Q}$  is the set of allowed  $q^*(f|\theta)$  defined by eq. (13).

3. **Update:** compute a new fraction of the approximate factor by dividing the new approximate posterior by the cavity,  $t_{n,\text{new}}^\alpha(\mathbf{u}) = q^*(f|\theta)/q^{\setminus n}(f|\theta)$ , and incorporate this fraction back in to obtain the updated factor,  $t_n(\mathbf{u}) = t_{n,\text{old}}^{1-\alpha}(\mathbf{u})t_{n,\text{new}}^\alpha(\mathbf{u})$ .

The above steps are iteratively repeated for each factor that needs to be approximated. Notice that the procedure only involves one likelihood factor to be handled at a time. In the case of analytically intractable likelihood functions, this often requires only low dimensional integrals to be computed. In other words, EP as transformed a high dimensional intractable integral that is hard to approximate into a set of low dimensional intractable integrals that are simpler to approximate. The procedure is not, in general guaranteed to converge but we did not observe any convergence issues in our experiments. Furthermore, it can be shown to be numerically stable when the factors are log-concave (as in GP regression and classification) (Seeger, 2008).

When  $\alpha = 1$ , Power EP is called EP and as  $\alpha \rightarrow 0$  the Power EP solution is the minimum of a variational free-energy. We will now show that these cases of Power EP recover FITC and Titsias's VFE solution respectively.

### 3.2 General Results for Gaussian Process Power EP

This section describes the Power EP steps in finer detail showing the complexity is  $\mathcal{O}(NM^2)$  and laying the ground work for the equivalence relationships. The appendix includes a full derivation.

We start by defining the approximate factors to be in natural parameter form, making it simple to combine and delete them,  $t_n(\mathbf{u}) = \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) = z_n \exp(\mathbf{u}^\top \mathbf{T}_{1,n} - \frac{1}{2} \mathbf{u}^\top \mathbf{T}_{2,n} \mathbf{u})$ . We consider full rank  $\mathbf{T}_{2,n}$ , but will show that the optimal form is rank 1. The parameterisation means the approximate posterior over the pseudo-points has natural parameters  $\mathbf{T}_{1,\mathbf{u}} = \sum_n \mathbf{T}_{1,n}$  and  $\mathbf{T}_{2,\mathbf{u}} = \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} + \sum_n \mathbf{T}_{2,n}$  inducing an approximate GP posterior with mean and covariance function,

$$m_{\mathbf{f}} = \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{T}_{2,\mathbf{u}}^{-1} \mathbf{T}_{1,\mathbf{u}}; \quad V_{\mathbf{f}\mathbf{f}'} = \mathbf{K}_{\mathbf{f}\mathbf{f}'} - \mathbf{Q}_{\mathbf{f}\mathbf{f}'} + \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{T}_{2,\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{f}'}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{f}'} \quad (15)$$

**Deletion:** The cavity for data point  $n$ ,  $q^{\setminus n}(f) \propto q^*(f)/t_n^\alpha(\mathbf{u})$ , has a similar form to the posterior, but the natural parameters are modified by the deletion step,  $\mathbf{T}_{1,\mathbf{u}}^{\setminus n} = \mathbf{T}_{1,\mathbf{u}} - \alpha \mathbf{T}_{1,n}$  and  $\mathbf{T}_{2,\mathbf{u}}^{\setminus n} = \mathbf{T}_{2,\mathbf{u}} - \alpha \mathbf{T}_{2,n}$ , yielding a new mean and covariance function

$$m_{\mathbf{f}}^{\setminus n} = \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{T}_{2,\mathbf{u}}^{\setminus n,-1} \mathbf{T}_{1,\mathbf{u}}^{\setminus n}; \quad V_{\mathbf{f}\mathbf{f}'}^{\setminus n} = \mathbf{K}_{\mathbf{f}\mathbf{f}'} - \mathbf{Q}_{\mathbf{f}\mathbf{f}'} + \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{T}_{2,\mathbf{u}}^{\setminus n,-1} \mathbf{K}_{\mathbf{u}\mathbf{f}'}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{f}'} \quad (16)$$

**Projection:** The central step in Power EP is the projection. Obtaining the new approximate unnormalised posterior  $q^*(f)$  by minimising  $\overline{\text{KL}}(\tilde{p}(f)||q^*(f))$  would naïvely appear intractable. Fortunately,

**Remark 1** *Because of the structure of the approximate posterior,  $q(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u})$ , the objective,  $\text{KL}(\tilde{p}(f)||q^*(f))$  is minimised when  $\mathbb{E}_{\tilde{p}(f)}[\phi(\mathbf{u})] = \mathbb{E}_{q^*(\mathbf{u})}[\phi(\mathbf{u})]$ , where  $\phi(\mathbf{u}) = [\mathbf{u}, \mathbf{u}\mathbf{u}^\top]$  are the sufficient statistics, that is when the moments at the pseudo-inputs are matched.*

This is the central result from which computational savings are derived. Furthermore, this moment matching condition would appear to necessitate computation of a set of integrals to find the zeroth, first and second moments. However, the technique known as ‘differentiation under the integral sign’ provides a useful shortcut that only requires one integral to compute the log-normaliser of the tilted distribution,  $\log \tilde{Z}_n = \log \mathbb{E}_{q^{\setminus n}(f)}[p^\alpha(y_n | \mathbf{f}_n)]$ , before differentiating w.r.t. the cavity mean to give

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n} \frac{d \log \tilde{Z}_n}{d m_{\mathbf{f}_n}^{\setminus n}}; \quad \mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n} \frac{d^2 \log \tilde{Z}_n}{d(m_{\mathbf{f}_n}^{\setminus n})^2} \mathbf{V}_{\mathbf{f}_n \mathbf{u}}^{\setminus n}. \quad (17)$$

**Update:** Having computed the new approximate posterior, the approximate factor  $t_{n,\text{new}}(\mathbf{u}) = q^*(f)/q^{\setminus n}(f)$  can be straightforwardly obtained, resulting in,

$$\mathbf{T}_{1,n,\text{new}} = \mathbf{V}_{\mathbf{u}}^{-1} \mathbf{m}_{\mathbf{u}} - (\mathbf{V}_{\mathbf{u}}^{\setminus n})^{-1} \mathbf{m}_{\mathbf{u}}^{\setminus n}, \quad \mathbf{T}_{2,n,\text{new}} = \mathbf{V}_{\mathbf{u}}^{-1} - (\mathbf{V}_{\mathbf{u}}^{\setminus n})^{-1}, \quad z_n^\alpha = \tilde{Z}_n e^{\mathcal{G}(q^{\setminus n}(\mathbf{u})) - \mathcal{G}(q^*(\mathbf{u}))},$$

where we have defined the log-normaliser  $\mathcal{G}(\tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)) = \log \int \tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2) d\mathbf{u}$ . Remarkably, these results and eqs. 17 reveals that  $\mathbf{T}_{2,n,\text{new}}$  is a rank-1 matrix. As a result, the minimal and simplest way to parameterise the approximate factor is  $t_n(\mathbf{u}) = z_n \mathcal{N}(\mathbf{K}_{\mathbf{f}_n \mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{u}; g_n, v_n)$ , where  $g_n$  and  $v_n$  are scalars, resulting in a significant memory saving and  $\mathcal{O}(NM^2)$  cost.

In addition to providing the approximate posterior after convergence, Power EP also provides an approximate log-marginal likelihood for model selection and hyper-parameter optimisation,

$$\log \mathcal{Z}_{\text{PEP}}(\theta) = \log \int p(f | \theta) \prod_n t_n(\mathbf{u}) d\mathbf{f} = \mathcal{G}(q^*(\mathbf{u})) - \mathcal{G}(p^*(\mathbf{u})) + \sum_n \log z_n. \quad (18)$$

Armed with these general results, we now consider the implications for Gaussian Process regression.

### 3.3 Gaussian Regression case

When the model contains Gaussian likelihood functions, closed-form expressions for the Power EP approximate factors at convergence can be obtained and hence the approximate posterior:

$$t_n(\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathbf{f}_n \mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{u}; y_n, \alpha D_{\mathbf{f}_n \mathbf{f}_n} + \sigma_y^2), \quad q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{K}_{\mathbf{u}\mathbf{f}} \bar{\mathbf{K}}_{\mathbf{f}\mathbf{f}}^{-1} \mathbf{y}, \mathbf{K}_{\mathbf{u}\mathbf{u}} - \mathbf{K}_{\mathbf{u}\mathbf{f}} \bar{\mathbf{K}}_{\mathbf{f}\mathbf{f}}^{-1} \mathbf{K}_{\mathbf{f}\mathbf{u}})$$

where  $\bar{\mathbf{K}}_{\mathbf{f}\mathbf{f}} = \mathbf{Q}_{\mathbf{f}\mathbf{f}} + \alpha \text{diag}(\mathbf{D}_{\mathbf{f}\mathbf{f}}) + \sigma_y^2 \mathbf{I}$  and  $\mathbf{D}_{\mathbf{f}\mathbf{f}} = \mathbf{K}_{\mathbf{f}\mathbf{f}} - \mathbf{Q}_{\mathbf{f}\mathbf{f}}$  as defined in section 2. These analytic expressions can be rigorously proven to be the stable fixed point of the Power EP procedure using theorem 1. Briefly, assuming the factors take the form above, the natural parameters of the cavity  $q^{\setminus n}(\mathbf{u})$  become,

$$\mathbf{T}_{1,\mathbf{u}}^{\setminus n} = \mathbf{T}_{1,\mathbf{u}} - \alpha \gamma_n y_n \mathbf{K}_{\mathbf{f}_n \mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}, \quad \mathbf{T}_{2,\mathbf{u}}^{\setminus n} = \mathbf{T}_{2,\mathbf{u}} - \alpha \gamma_n \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{f}_n} \mathbf{K}_{\mathbf{f}_n \mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}, \quad (19)$$

where  $\gamma_n^{-1} = \alpha D_{\mathbf{f}_n \mathbf{f}_n} + \sigma_y^2$ . The subtracted quantities in the equations above are exactly the contribution the likelihood factor makes to the cavity distribution (see theorem 1)

so  $\int q^n(f) p^\alpha(y_n | f_n) df_{\neq \mathbf{u}} = q^n(\mathbf{u}) \int p(f_n | \mathbf{u}) p^\alpha(y_n | f_n) df_n \propto q(\mathbf{u})$ . Therefore, the posterior approximation remains unchanged after an update and the form for the factors above is the fixed point. Moreover, the approximate log-marginal likelihood is also analytically tractable,

$$\log \mathcal{Z}_{\text{PEP}} = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{K}_{\mathbf{ff}}| - \frac{1}{2} \mathbf{y}^\top \mathbf{K}_{\mathbf{ff}}^{-1} \mathbf{y} - \frac{1-\alpha}{2\alpha} \sum_n \log(1 + \alpha D_{f_n f_n} / \sigma_y^2).$$

We now look at special cases and the correspondence to the methods discussed in section 2.

**Remark 2** When  $\alpha = 1$  [EP], the Power EP posterior becomes the FITC posterior and the Power EP approximate marginal likelihood becomes the FITC marginal likelihood. In other words, the FITC approximation for GP regression is, surprisingly, equivalent to running an EP algorithm for sparse GP posterior approximation to convergence.

**Remark 3** As  $\alpha \rightarrow 0$  the approximate posterior and approximate marginal likelihood are identical to that of the VFE approach (Titsias, 2009b). This result uses the limit:  $\lim_{x \rightarrow 0} x^{-1} \log(1+x) = 1$ . So FITC and Titsias’s VFE approach employ the same form of pseudo-point approximation, but refine it in different ways.

### 3.4 Extensions: structured, inter-domain and multi-power Power EP approximations

The framework can now be generalised in three orthogonal directions:

1. enable structured approximations to be handled that retain more dependencies in the spirit of PITC (see section 2.1)
2. incorporate inter-domain pseudo-points thereby adding further flexibility to the form of the approximate posterior
3. employ different powers  $\alpha$  for each factor (thereby enabling e.g. VFE updates to be used for some data points and EP for others).

Given the groundwork above, these three extensions are straightforward. In order to handle structured approximations, we take inspiration from PITC and partition the data into  $B$  disjoint blocks  $\mathbf{y}_b = \{y_n\}_{n \in \mathcal{B}_b}$  (see section 2.1). Each PEP factor update will then approximate an entire block which will contain a set of data points, rather than just a single one. This is a style of EP approximation that has recently been used to distribute Monte Carlo algorithms across many machines (Gelman et al., 2014; Xu et al., 2014).

In order to handle inter-domain variables, we define a new domain via a linear transform  $g(\mathbf{x}) = \int d\mathbf{x}' W(\mathbf{x}, \mathbf{x}') f(\mathbf{x}')$  which now contains the pseudo-points  $g = \{g_{\neq \mathbf{u}}, \mathbf{u}\}$ . Choices for  $W(\mathbf{x}, \mathbf{x}')$  include Gaussians or wavelets. These two extensions mean that the approximation becomes,

$$p(f, g | \theta) \prod_b p(\mathbf{y}_b | f, \theta) \approx p(f, g | \theta) \prod_b t_b(\mathbf{u}) = q^*(f | \theta). \quad (20)$$

Power EP is then performed using private powers  $\alpha_b$  for each data block, which is the third generalisation mentioned above. Analytic solutions are again available (covariance matrices now incorporate the inter-domain transform)

$$t_b(\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathbf{f}_b\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathbf{y}_b, \alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b} + \sigma_y^2\mathbf{I}), \quad q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{K}_{\mathbf{uf}}\bar{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uf}}\bar{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{K}_{\mathbf{fu}})$$

where  $\bar{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \text{blkdiag}(\{\alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b}\}_{b=1}^B) + \sigma_y^2\mathbf{I}$  and  $\text{blkdiag}$  builds a block-diagonal matrix from its inputs. The approximate log-marginal likelihood can also be obtained in closed-form,

$$\log \mathcal{Z}_{\text{PEP}} = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\bar{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2} \mathbf{y}^\top \bar{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{y} + \sum_b \frac{1 - \alpha_b}{2\alpha_b} \log (\mathbf{I} + \alpha_b \mathbf{D}_{\mathbf{f}_b\mathbf{f}_b} / \sigma_y^2).$$

**Remark 4** When  $\alpha_b = 1$  and  $W(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$  the structured Power EP posterior becomes the PITC posterior and the Power EP approximate marginal likelihood becomes the PITC marginal likelihood. Additionally, when  $B = N$  we recover FITC as discussed in section 3.3.

**Remark 5** When  $\alpha_b \rightarrow 0$  and  $W(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$  the structured Power EP posterior and approximate marginal likelihood becomes identical to the VFE approach (Titsias, 2009b). (See fig. 1 for more relationships.)

### 3.5 Classification

For classification, the non-Gaussian likelihood prevents an analytic solution. As such, the iterative Power EP procedure is required to obtain the approximate posterior. The projection step requires computation of the log-normaliser of the tilted distribution,  $\log \tilde{Z}_n = \log \mathbb{E}_{q \setminus n(f)}[p^\alpha(y_n|f)] = \log \mathbb{E}_{q \setminus n(\mathbf{f}_n)}[\Phi^\alpha(y_n \mathbf{f}_n)]$ . For general  $\alpha$ , this quantity is not available in closed form<sup>2</sup>. However, it is a one-dimensional Gaussian integral and can be approximated using Gauss-Hermite quadrature, resulting in an approximate update for the posterior mean and covariance. The approximate log-marginal likelihood can also be obtained and used for hyper-parameter optimisation. As  $\alpha \rightarrow 0$ , it becomes the variational free-energy used in (Hensman et al., 2015) which employs quadrature for the same purpose. These relationships are shown in fig. 1 which also shows that inter-domain transformations and structured approximations have not yet been employed in the classification setting. In our view, the inter-domain generalisation would be a sensible one to pursue and it is mathematically and algorithmically straightforward. The structured approximation variant is more complicated as it requires multiple non-linear likelihoods to be handled at each step of EP. This will require further approximation such as using Monte Carlo methods (Gelman et al., 2014; Xu et al., 2014).

Since the proposed Power EP approach is general, an extension to other likelihood functions is as simple as for VFE methods (Dezfouli and Bonilla, 2015). For example, the multinomial probit likelihood can be handled in the same way as the binary case, where the

2. except for special cases, e.g. when  $\alpha = 1$  and  $\Phi(x)$  is the probit inverse link function,  $\Phi(x) = \int_{-\infty}^x \mathcal{N}(a; 0, 1) da$ .

log-normaliser of the tilted distribution can be computed using a  $C$ -dimensional Gaussian quadrature [ $C$  is the number of classes] (Seeger and Jordan, 2004) or nested EP (Riihimäki et al., 2013).

### 3.6 Complexity

The computational complexity of all the regression and classification methods described in this section is  $\mathcal{O}(NM^2)$  for training, and  $\mathcal{O}(M^2)$  per test point for prediction. The training cost can be further reduced to  $\mathcal{O}(M^3)$ , in a similar vein to the uncollapsed VFE approach (Hensman et al., 2013, 2015), by employing stochastic updates of the posterior and stochastic optimisation of the hyper-parameters using minibatches of data points (Hernández-Lobato and Hernández-Lobato, 2016). In particular, the Power-EP update steps in section 3.1 are repeated for only a small subset of training points and for only a small number of iterations. The approximate log-marginal likelihood in eq. (18) is then computed using this minibatch and optimised as if the Power-EP procedure has converged. This approach results in a computationally efficient training scheme, at the cost of returning noisy hyper-parameter gradients. In practice, we find that the noise can be handled using stochastic optimisers such as Adam (Kingma and Ba, 2015). In summary, given these advances the general PEP framework is as scalable as variational inference.

## 4. Gaussian Process State-space Model

The Power EP framework can be generalised to models with latent variables such as the GP latent variable model (Lawrence, 2005) or the GP state space model (GPSSM) (Wang et al., 2005). Here we focus on the GPSSM for brevity. The GPSSM contains continuous valued latent variables  $\mathbf{x}$  that evolve according to nonlinear dynamics with Gaussian innovations noise and observations  $\mathbf{y}$  that are Gaussian conditioned on the latents,

$$p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; f(\mathbf{x}_{t-1}), \sigma_x^2 \mathbf{I}), \quad p(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t; \mathbf{C}\mathbf{x}_t, \mathbf{R}_y).$$

The exact and intractable posterior over the latent function  $f$  and the latent variables  $\mathbf{x}$  is  $p(\mathbf{x}, f|\mathbf{y}) \propto p(\mathbf{x}_0)p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u}) \prod_t p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) \prod_t p(\mathbf{y}_t|\mathbf{x}_t)$ . We posit the following approximate posterior,  $q(\mathbf{x}, f) \propto p(\mathbf{x}_0)p(f_{\neq \mathbf{u}}|\mathbf{u}) \prod_t \phi_t(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u}) \prod_t p(\mathbf{y}_t|\mathbf{x}_t)$  and we use a factored approximation  $\phi_t(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u}) = \lambda_t(\mathbf{x}_{t-1})\beta_t(\mathbf{x}_t)\gamma_t(\mathbf{u})$ . The Power EP procedure updates  $\phi_t(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u})$  using the steps described in section 3. One crucial difference is the use of a Gaussian projection nested in the computation of the log-normaliser of the tilted distribution,  $\log \tilde{Z}_t$  (see appendices). In spite of this Gaussian projection, as  $\alpha \rightarrow 0$ , we recover the exact variational free-energy treatment. In addition, if an additional factorised assumption over  $\mathbf{x}_{1:T}$ ,  $q(\mathbf{x}_{1:T}) = \prod_t q(\mathbf{x}_t)$  is introduced we recover the variational mean field approximation used by McHutchon (2014).

## 5. Experiments

The general framework described above lays out a large space of potential inference algorithms suggesting many exciting directions for innovation. The experiments considered in the paper will investigate only one aspect of this space; how do algorithms that are intermediate between VFE ( $\alpha = 0$ ) and EP/FITC ( $\alpha = 1$ ) perform? Specifically, we will investigate

how the performance of the inference scheme varies as a function of  $\alpha$  and whether this depends on; the type of problem (classification, regression or state-space modelling); the dataset (synthetic datasets, 8 real world regression datasets, 6 classification datasets, and one time-series dataset are considered); the performance metric (we compare metrics that require point-estimates to those that are uncertainty sensitive). An important by-product of the experiments is that they provide a comprehensive comparison between the VFE and EP approaches which has been an important area of debate in its own right.

The results presented below are compact summaries of a large number of experiments full details of which are included in the appendix (along with additional experiments). Python and Matlab implementations are available at [http://github.com/thangbui/sparseGP\\_powerEP](http://github.com/thangbui/sparseGP_powerEP).

### 5.1 Regression on Synthetic Datasets

In the first experiment, we investigate the performance of the proposed Power-EP method on toy regression datasets where ground truth is known. We vary  $\alpha$  (from 0 VFE to 1 EP/FITC) and the number of pseudo-points (from 5 to 500). We use thirty datasets, each comprising 1000 data points that were drawn from a GP with an Automatic Relevance Determination squared exponential kernel. A 50:50 train/test split was used. The hyper-parameters and pseudo-inputs were optimised using L-BFGS with a maximum of 2000 function evaluations. The performances are compared using two metrics: standardised mean squared error (SMSE) and standardised mean log loss (SMLL) as described in (Rasmussen and Williams, 2005). The approximate negative log-marginal likelihood (NLML) for each experiment is also computed. The mean performance using Power EP with different  $\alpha$  values and full GP regression is shown in fig. 3. The results demonstrate that as  $M$  increases, the SMLL and SMSE of the sparse methods approach that of full GP. Power EP with  $\alpha = 0.8$  or  $\alpha = 1$  (EP) overestimates the log-marginal likelihood, even for a modest number of pseudo-points. Importantly, however, an intermediate value of  $\alpha$  in the range 0.5-0.8 seems to be best for prediction on average, outperforming both EP and VFE.

### 5.2 Regression on Real-world Datasets

The experiment above was replicated on 8 UCI regression datasets, each with 20 train/test splits. We varied  $\alpha$  between 0 and 1, and  $M$  was varied between 5 and 200. Full details of the experiments along with extensive additional analysis is presented in the appendices. Here we concentrate on several key aspects. First we consider pairwise comparisons between VFE ( $\alpha \rightarrow 0$ ), Power-EP with  $\alpha = 0.5$  and EP/FITC ( $\alpha = 1$ ) on both the SMSE and SMLL evaluation metrics. Power-EP with  $\alpha = 0.5$  was chosen because it is the mid-point between VFE and EP and because settings around this value empirically performed the best on average across all datasets, splits, numbers of inducing points, and evaluation metrics.

In fig. 4A we plot (for each dataset, each split and each setting of  $M$ ) the evaluation scores obtained using one inference algorithm (e.g. PEP  $\alpha = 0.5$ ) against the score obtained using another (e.g. VFE  $\alpha = 0$ ). In this way, points falling below the identity line indicate experiments where the method on the y-axis outperformed the method on the x-axis. These results have been collapsed by forming histograms of the difference in the performance of the two algorithms, such that mass to the right of zero indicates the method on the y-axis



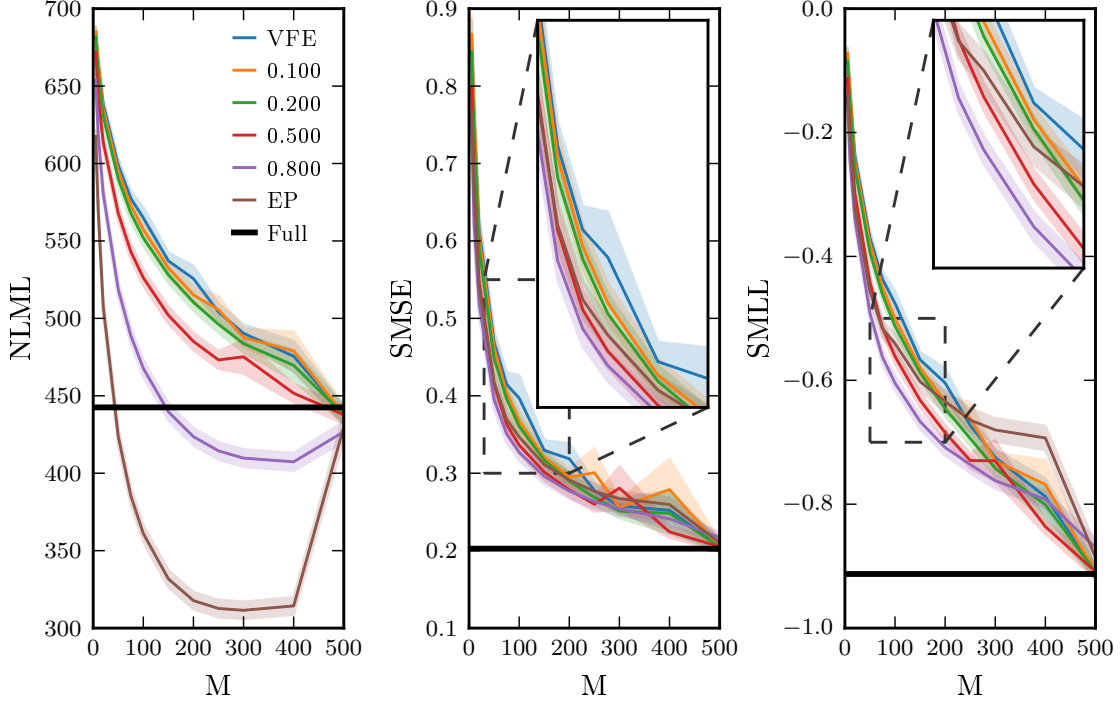


Figure 3: The performance of various  $\alpha$  values averaged over 30 trials. See text for more details

outperformed that on the x-axis. The proportion of mass on each side of the histogram, also indicated on the plots, shows in what fraction of experiments one method returns a more accurate result than the other. This is a useful summary statistic, linearly related to the average rank, that we will use to unpack the results. The average rank is insensitive to the magnitude of the performance differences and readers might worry that this might give an overly favourable view of a method that performs the best frequently, but only by a tiny margin, and when it fails it does so catastrophically. However, the histograms indicate that the methods that win most frequently tend also to ‘win big’ and ‘lose small’, although EP is a possible exception to this trend (see the outliers below the identity line on the bottom right-hand plot).

A clear pattern emerges from these plots. First PEP  $\alpha = 0.5$  is the best performing approach on the SMSE metric, outperforming VFE 67% of the time and EP 78% of the time. VFE is better than EP on the SMSE metric 64% of the time. Second, EP performs the best on the SMLL metric, outperforming VFE 93% of the time and PEP  $\alpha = 0.5$  71% of the time. PEP  $\alpha = 0.5$  outperforms VFE in terms of the SMLL metric 93% of the time.

These pairwise rank comparisons have been extended to other values of  $\alpha$  in figure Figure 5A. Here, each row of the figure compares one approximation with all others. Horizontal bars indicate that the methods have equal average rank. Upward sloping bars indicate the method shown on that row has lower average rank (better performance), and downward sloping bars indicate higher average rank (worse performance). The plots show that PEP  $\alpha = 0.5$  outperforms all other methods on the SMSE metric, except for PEP  $\alpha = 0.6$  which

is marginally better. EP is outperformed by all other methods, and VFE only outperforms EP on this metric. On the other hand, EP is the clear winner on the SMLL metric, with performance monotonically decreasing with  $\alpha$  so that VFE is the worst.

The same pattern of results is seen when we simultaneously compare all of the methods, rather than considering sets of pairwise comparisons. The average rank plots shown in fig. 4B were produced by sorting the performances of the 8 different approximating methods for each dataset, split, and number of pseudo-points  $M$  and assigning a rank. These ranks are then averaged over all datasets and their splits, and settings of  $M$ . PEP  $\alpha = 0.5$  is the best for the SMSE metric, and the two worst methods are EP and VFE. PEP  $\alpha = 0.8$  is the best for the SMLL metric, with EP and PEP  $\alpha = 0.6$  not far behind (when EP performs poorly it can do so with a large magnitude, explaining the discrepancy with the pairwise ranks).

There is some variability between individual datasets, but the same general trends are clear: For MSE  $\alpha = 0.5$  is better than VFE on 6/8 datasets and EP on 8/8 datasets, whilst VFE is better than EP on 3 datasets (the difference on the others being small). For NLL EP is better than  $\alpha = 0.5$  on 5/8 datasets and VFE on 7/8 datasets, whilst  $\alpha = 0.5$  is better than VFE on 8/8 datasets. Performance tends to increase for all methods as a function of the number of pseudo-points  $M$ . The interaction between the choice of  $M$  and the best performing inference method is often complex and variable across datasets making it hard to give precise advice about selecting  $\alpha$  in an  $M$  dependent way.

In summary, we make the following recommendations based on these results for GP regression problems. For a MSE loss, we recommend using  $\alpha = 0.5$ . For a NLL we recommend using EP. It is possible that more fine grained recommendations are possible based upon details of the dataset and the computational resources available for processing, but further work will be needed to establish this.

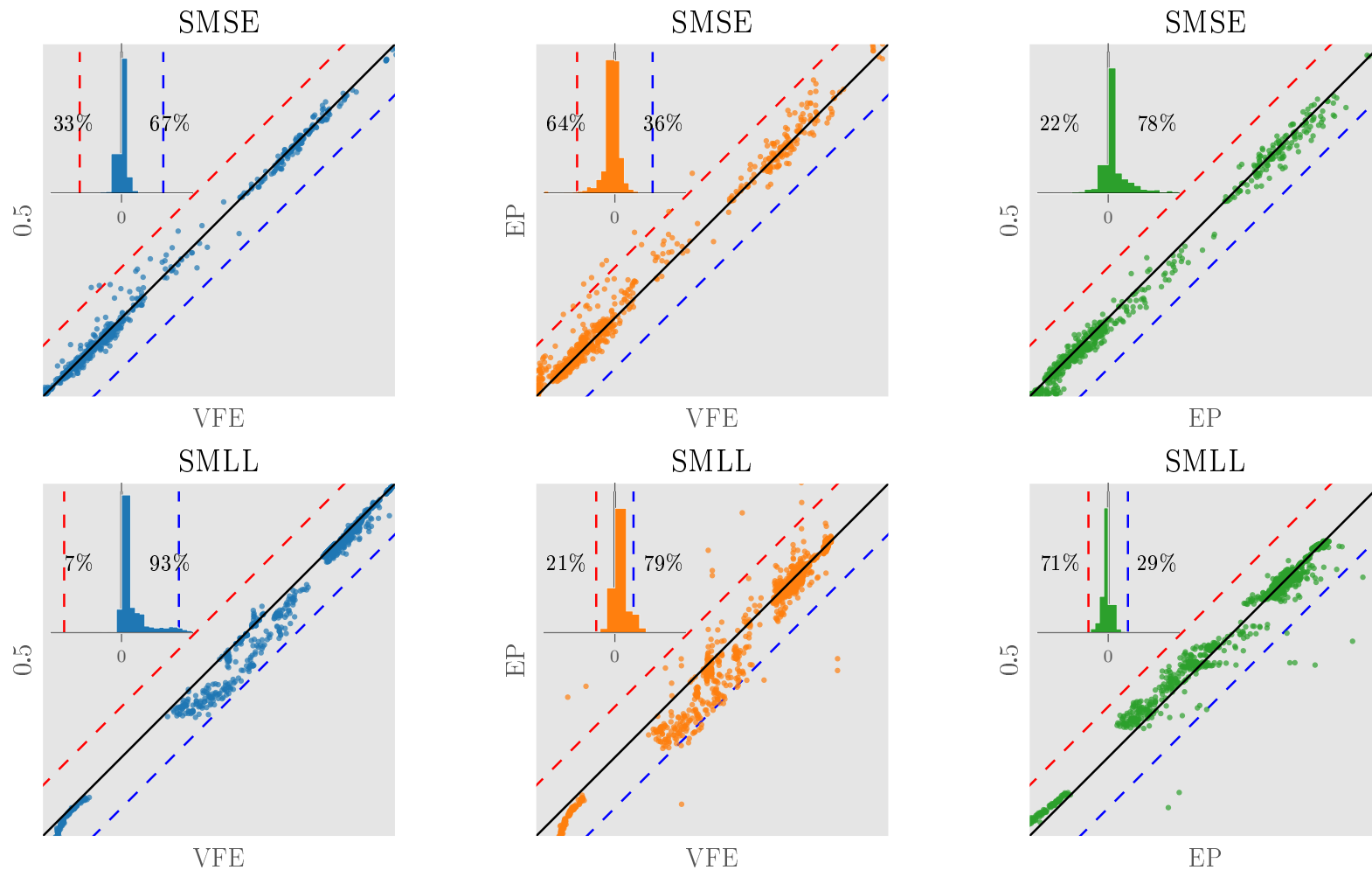


Figure 4: Pair-wise comparisons between Power-EP with  $\alpha = 0.5$ , EP ( $\alpha = 1$ ) and VFE ( $\alpha \rightarrow 0$ ), evaluated on several regression datasets and various settings of  $M$ . Each coloured point is the result for one split. Points that are below the diagonal line illustrate the method on the  $y$ -axis is better than the method on the  $x$ -axis. The inset diagrams show the histograms of the difference between methods ( $x$ -value  $- y$ -value), and the counts of negative and positive differences. Note that this indicates pairwise ranking of the two methods. Positive differences mean the  $y$ -axis method is better than the  $x$ -axis method and vice versa. For example, the middle, bottom plot shows EP is on average better than VFE.

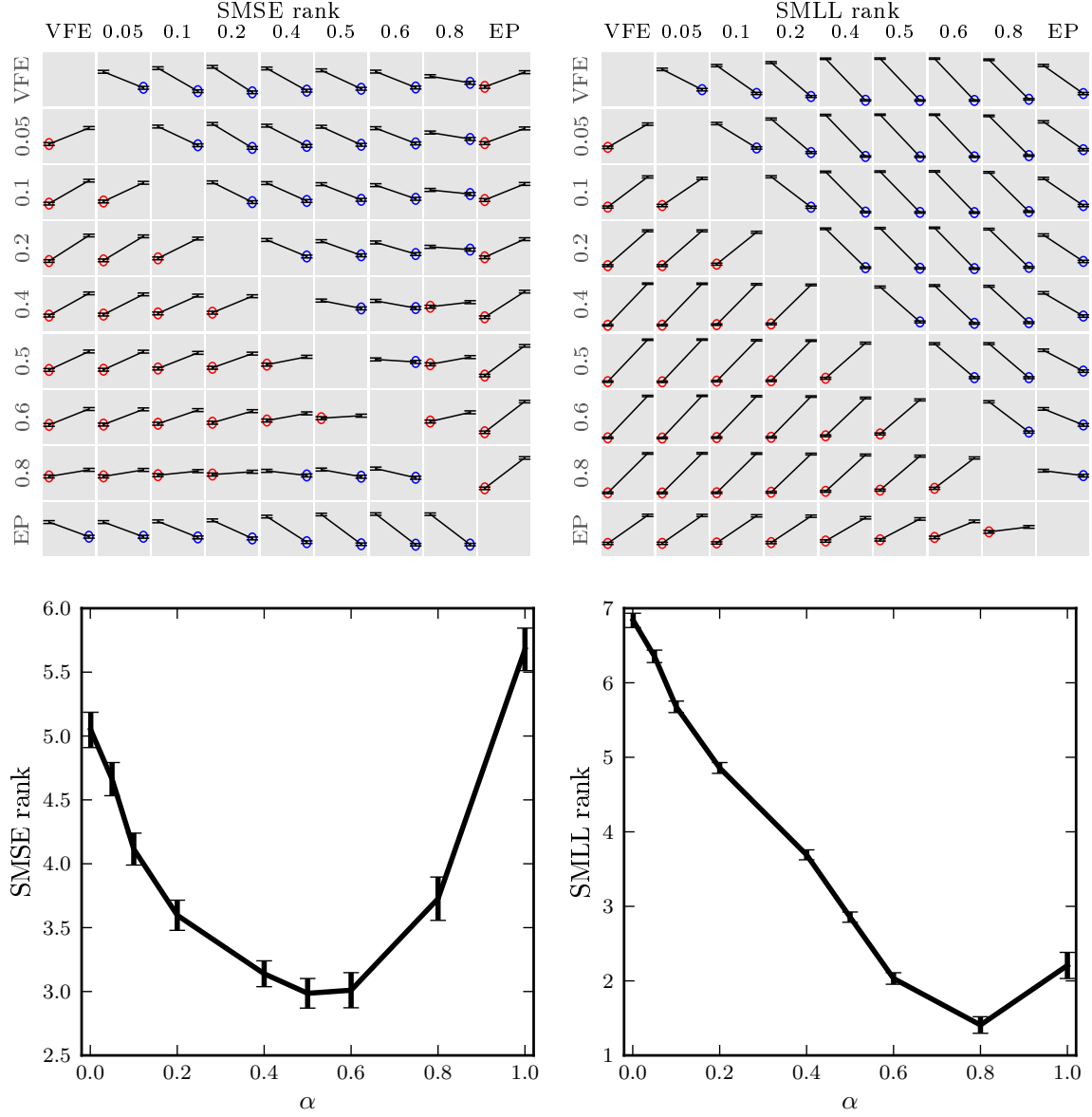


Figure 5: Average ranking of various  $\alpha$  values in the regression experiment, lower is better. Top plots show the pairwise comparisons. Red circles denote rows being better than the corresponding columns, and blue circles mean vice versa. Bottom plots show the ranks of all methods when being compared together. Intermediate  $\alpha$  values (not EP or VFE) are best on average.

### 5.3 Binary Classification

We also evaluated the Power EP method on 6 UCI classification datasets, each has 20 train/test splits. Again  $\alpha$  was varied between 0 and 1, and  $M$  was varied between 10 and 100. We adopt the experimental protocol discussed in section 3.6, including: (i) not waiting

for Power EP to converge before making hyper-parameter updates, (ii) using minibatches of data points for each Power EP sweep, (iii) parallel factor updates. The Adam optimiser was used with default hyper-parameters to handle the noisy gradients produced by these approximations (Kingma and Ba, 2015). Similar to the regression experiment, we compare the methods using the pairwise ranking plots on the test error and negative log-likelihood (NLL) evaluation metrics.

In fig. 6, we plot (for each dataset, each split and each setting of  $M$ ) the evaluation scores using one inference algorithm against the score obtained using another [see section 5.2 for a detailed explanation of the plots]. The test error results show that PEP  $\alpha = 0.5$  and EP outperforms VFE with both beating VFE 91% of the time. PEP  $\alpha = 0.5$  is marginally better than EP in this metric. Similar observations can be made using the NLL metric: PEP  $\alpha = 0.5$  and EP are better than VFE with both beating VFE 99% of the time, and PEP  $\alpha = 0.5$  slightly outperforms EP (62% of the time vs. 38%).

We repeat the pairwise comparison above to all methods and show the results in fig. 7. The plots show PEP  $\alpha = 0.5$  and PEP  $\alpha = 0.6$  are best performing methods on both test error and NLL metrics. PEP  $\alpha = 0.4$  and  $\alpha = 0.8$  are the next close competitors. VFE is notably outperformed by all other methods. Similar to the regression experiment, we observe the same pattern of results when all methods are simultaneously compared, as shown in fig. 7.

There is some variability between individual datasets, but the general trends are clear and consistent with the pattern noted above. For test error, PEP  $\alpha = 0.5$  is better than VFE on 6/6 datasets and PEP is better than EP on 2/6 datasets (the differences on the other 4 datasets are small). EP is also better than VFE on all datasets. For NLL, PEP  $\alpha = 0.5$  is better than VFE on 6/6 datasets and PEP is better than EP on 3/6 datasets (EP is better on 2 and one dataset has no clear winner). EP is also better than VFE on all datasets. The finding that PEP is slightly better than EP on the NLL metric is surprising as we expected EP perform the best on the uncertainty sensitive metric (just as was discovered in the regression case). The full results are included in the appendices (see figs 25, 26 and 27). Similar to the regression case, we observe that as  $M$  increases, the performance tends to be better for all methods and the differences between the methods tend to become smaller, but we have not found evidence for systematic sensitivity to the nature of the approximation.

In summary, we make the following recommendations based on these results for GP classification problems. For a raw test error loss and for NLL, we recommend using  $\alpha = 0.5$  (or  $\alpha = 0.6$ ). It is possible that more fine grained recommendations are possible based upon details of the dataset and the computational resources available for processing, but further work will be needed to establish this.

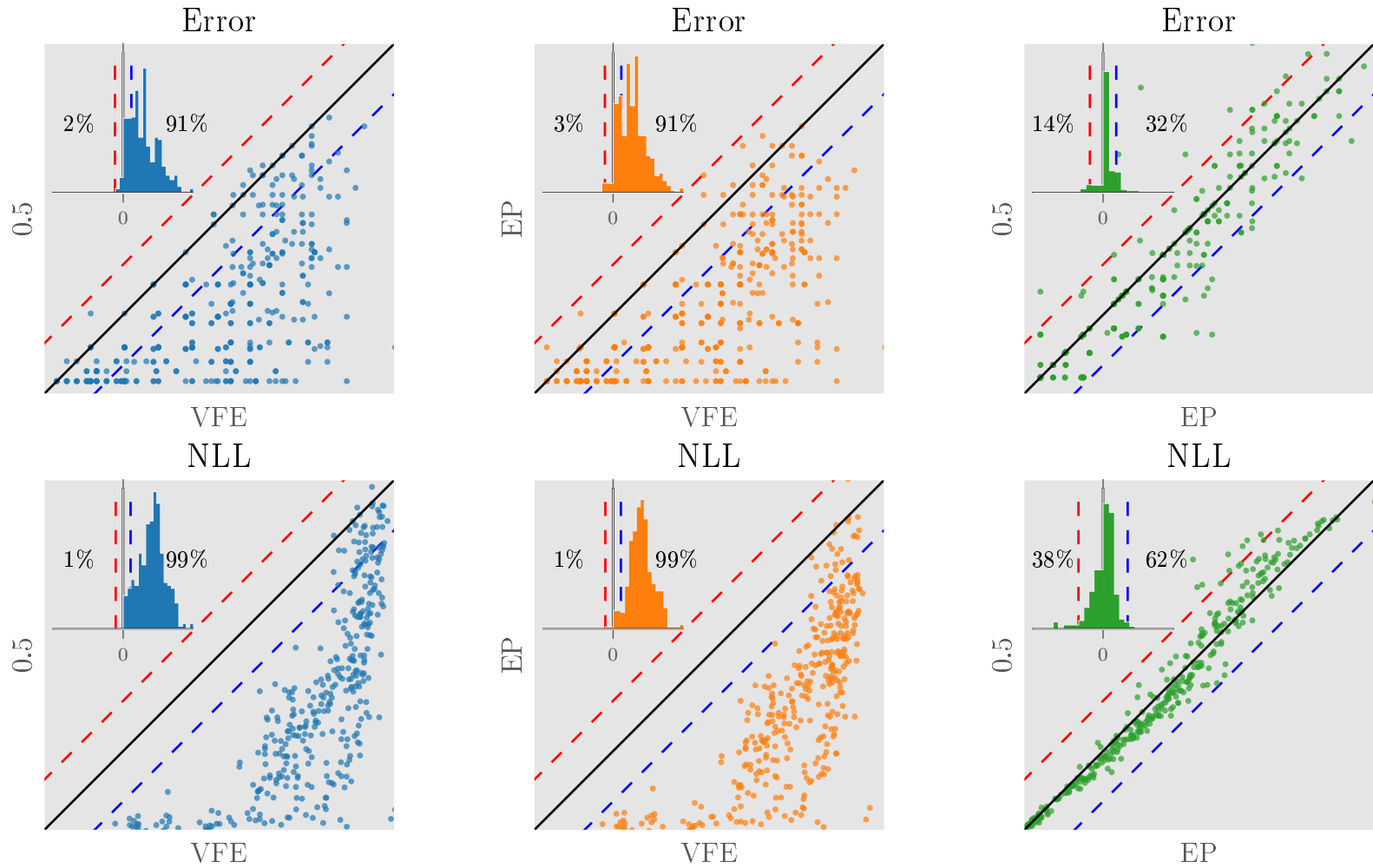


Figure 6: Pair-wise comparisons between Power-EP with  $\alpha = 0.5$ , EP ( $\alpha = 1$ ) and VFE ( $\alpha \rightarrow 0$ ), evaluated on several classification datasets and various settings of  $M$ . Each coloured point is the result for one split. Points that are below the diagonal line illustrate the method on the  $y$ -axis is better than the method on the  $x$ -axis. The inset diagrams show the histograms of the difference between methods ( $x$ -value -  $y$ -value), and the counts of negative and positive differences. Note that this indicates pairwise ranking of the two methods. Positive differences means the  $y$ -axis method is better than the  $x$ -axis method and vice versa.

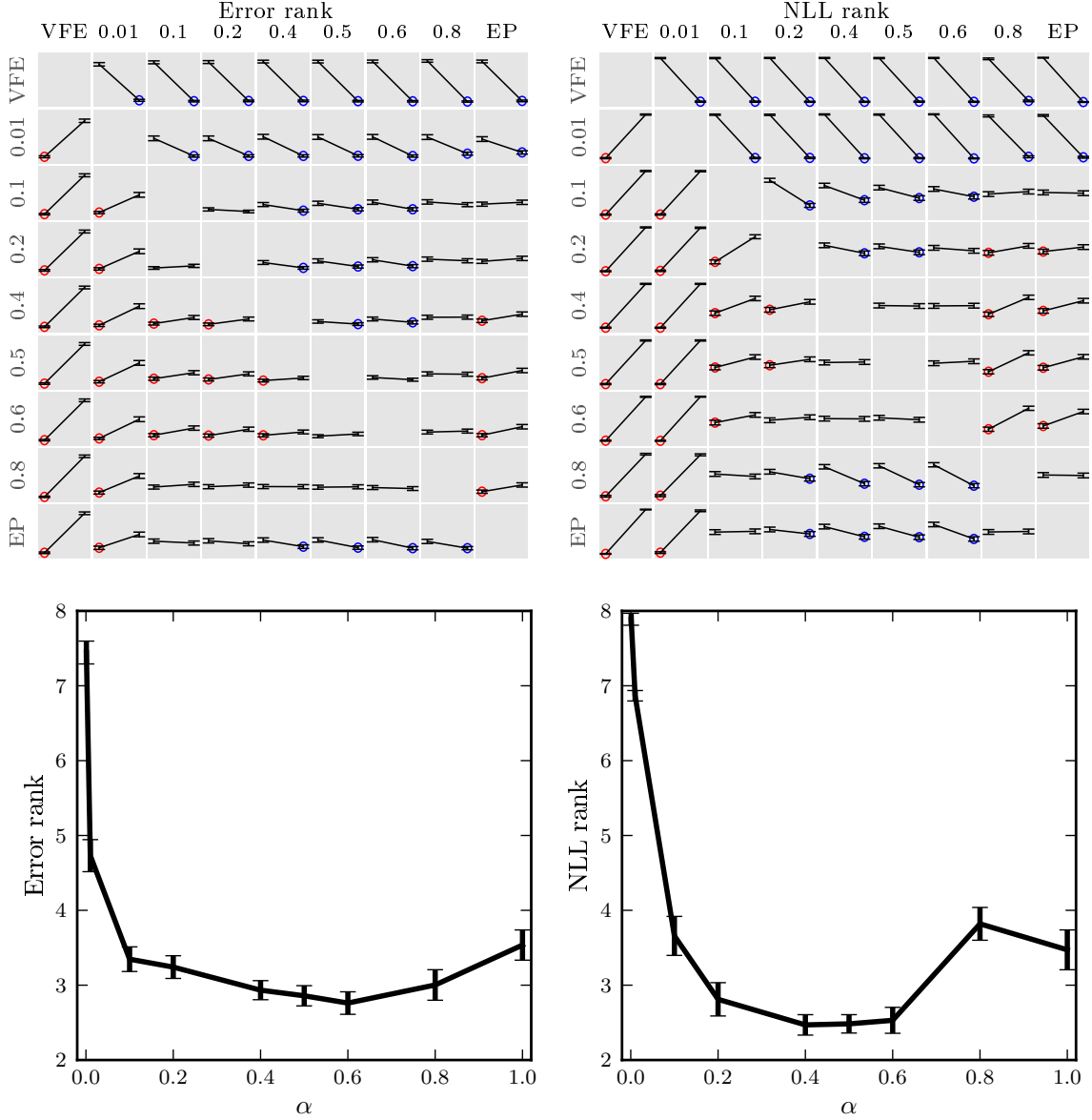


Figure 7: Average ranking of various  $\alpha$  values in the classification experiment, lower is better. Top plots show the pairwise comparisons. Red circles denote rows being better than the corresponding columns, and blue circles mean vice versa. Bottom plots show the ranks of all methods when being compared together. Intermediate  $\alpha$  values (not EP or VFE) are best on average.

#### 5.4 Learning Gaussian Process State-space Models

Finally, we evaluate the Power EP approach for the GPSSM, using a one-dimensional non-linear system governed by  $p(x_t|x_{t-1}) = \mathcal{N}(x_t; f(x_t), 1)$  and  $p(y_t|x_t) = \mathcal{N}(y_t; x_t, 1)$  where the transition function is  $f_{x_t} = x_t + 1$  if  $x_t < 4$  and  $-4x_t + 21$  if  $x_t > 4$ . Figure 8 shows the

transition function and associated uncertainty learnt using factored EP ( $\alpha = 1$ ) and two different VFE approximations ( $\alpha \rightarrow 0$ ): factored  $q(\mathbf{x}_{1:T}) = \prod_t q(\mathbf{x}_t)$  and McHutchon’s Markov approximation  $q(\mathbf{x}_{1:T}) = \prod_t q(\mathbf{x}_t | \mathbf{x}_{t-1})$  (Frigola et al., 2014; McHutchon, 2014). While EP accurately captures the underlying dynamics, factored VFE prefers a simple and inaccurate transition function. This failure mode of VFE methods for time series is well-documented (Turner and Sahani, 2011) and is due to the bias introduced by the KL divergence term, which means the variational algorithm will tend to learn simpler functions since then the true posterior over  $\mathbf{x}_{1:T}$  is more Gaussian and less coupled. The Markov VFE approach fixes this issue by introducing dependencies in the approximate posterior over  $\mathbf{x}$ , effectively removing this bias, but introducing greater computational cost.

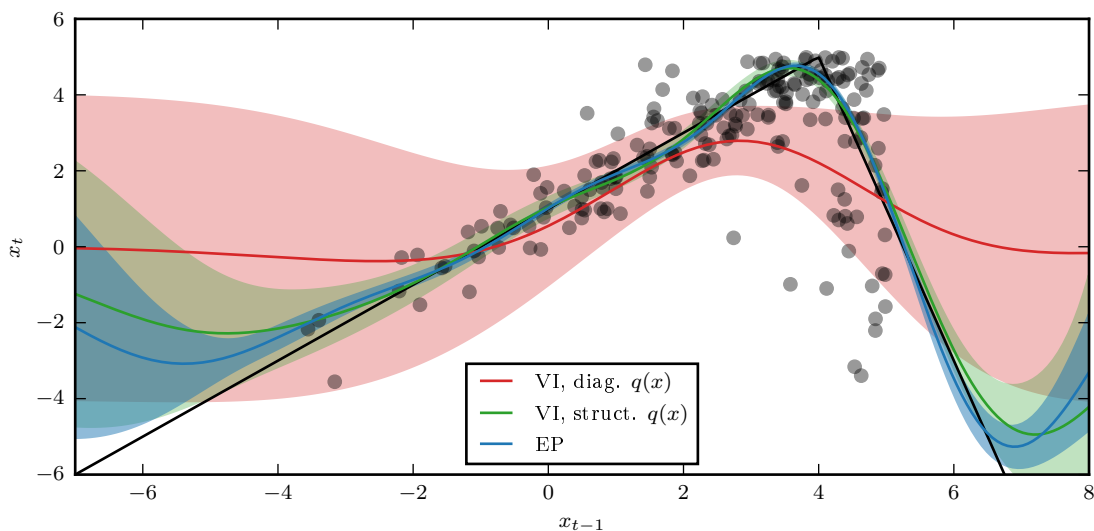


Figure 8: Learnt transition function and its uncertainty using EP and VI vs. ground truth [black]. Noisy previous/current hidden states are shown as black cross.

## 6. Discussion

The results presented above employed (approximate) type-II maximum likelihood fitting of the hyper-parameters. This estimation method is known in some circumstances to overfit the data. It is therefore conceivable therefore that pseudo-point approximations, which have a tendency to encourage under-fitting due to their limited representational capacity, could be beneficial due to them mitigating overfitting. We do not believe that this is a strong effect in the experiments above. For example, in the synthetic data experiments the NLML, SMSE and NMLL obtained from fitting the unapproximated GP were similar to those obtained using the GP from which the data were generated, indicating that overfitting is not a strong effect (see figure 9 in the appendix). It is true that EP and  $\alpha = 0.8$  overestimates the marginal likelihood in the synthetic data experiments, but this is a distinct effect from over-fitting which would, for example, result in overconfident predictions on the



test dataset. The SMSE and SMLL on the training and test sets, for example, are similar which is indicative of a well-fit model.

One of the features of the approximate generative models introduced in section 2.1 for regression, is that they contain input-dependent noise, unlike the original model. Many datasets contain noise of this sort and so approximate models like FITC and PITC are arguably more appropriate than the original unapproximated regression model (Snelson, 2007). Motivated by this train of reasoning, Titsias (2009a) applied the variational free-energy approximation to the FITC generative model an approach that was later generalised by Hoang et al. (2016) to encompass a more general class of input dependent noise, including Markov structure (Low et al., 2015). Here the insight is that the resulting variational lower bound separates over data points (Hensman et al., 2013) and is, therefore, amenable to stochastic optimisation using minibatches unlike the marginal likelihood. In a sense, these approaches unify the approximate generative modelling approach, including the FITC and PITC variants, with the variational free-energy methods. Indeed, one approach is to posit the desired form of the optimal variational posterior, and to work backwards from this to construct the generative model implied (Hoang et al., 2015). However, these approaches are quite different from the one described in this paper where FITC and PITC are shown to emerge in the context of approximating the original unapproximated GP regression model using Power EP. Indeed, if the goal really is to model input dependent noise, it is not at all clear that generative models like FITC are the most sensible. For example, FITC uses a single set of hyper-parameters to describe the variation of the underlying function and the input dependent noise.

## 7. Conclusion

This paper provided a new unifying framework for GP pseudo-point approximations based on Power EP that subsumes many previous approaches including FITC, PITC, DTC, Titsias’s VFE method, Qi et al’s EP method, and inter-domain variants. It provided a clean computational perspective on the seminal work of Csató and Opper that related FITC to EP, before extending their analysis significantly to include a closed form Power EP marginal likelihood approximation for regression, connections to PITC, and further results on classification and GPSSMs. The new framework was used to devise new algorithms for GP regression, GP classification and GPSSMs. Extensive experiments indicate that intermediate values of Power EP with the power parameter set to  $\alpha = 0.5$  often outperform the state-of-the-art EP and VFE approaches. The new framework suggests many interesting directions for future work in this area that we have not explored, for example, extensions to online inference, combinations with special structured matrices (e.g. circulant and Kronecker structure), Bayesian hyper-parameter learning, and applications to richer models. The current work has only scratched the surface, but we believe that the new framework will form a useful theoretical foundation for the next generation of GP approximation schemes.

## Acknowledgments

The authors would like to thank Prof. Carl Edward Rasmussen, Niles Tripuraneni, Matthias Bauer, and Hugh Salimbeni for insightful comments and discussion. TDB thanks Google for funding his European Doctoral Fellowship. RET thanks EPSRC grants EP/G050821/1 and EP/L000776/1.

## Appendix A. Some relevant linear algebra and function expansion identities

In this section, we include some identities that will be used throughout the following sections.

The Woodbury matrix identity or Woodbury formula is:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}. \quad (21)$$

In general,  $C$  need not be invertible, we can use the Binomial inverse theorem,

$$(A + UCV)^{-1} = A^{-1} - A^{-1}UC(C + CVA^{-1}UC)^{-1}CVA^{-1}. \quad (22)$$

When  $C$  is an identity matrix and  $U$  and  $V$  are vectors, the Woodbury identity can be shortened and become the Sherman-Morrison formula,

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u}. \quad (23)$$

Another useful identity is the matrix determinant lemma,

$$\det(A + uv^T) = (1 + v^T A^{-1}u)\det(A). \quad (24)$$

The above theorem can be extend for matrices  $U$  and  $V$ ,

$$\det(A + UV^T) = \det(I + V^T A^{-1}U)\det(A). \quad (25)$$

We also make use of the following Maclaurin series,

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \quad (26)$$

$$\text{and } \log(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \cdots. \quad (27)$$

## Appendix B. KL minimisation between Gaussian processes and moment matching

The difficult step of Power-EP is the projection step, that is how to find the posterior approximation  $q(f)$  that minimises the KL divergence,  $\text{KL}(\tilde{p}(f)||q(f))$ , where  $\tilde{p}(f)$  is the tilted distribution. We have chosen the form of the approximate posterior

$$q(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u})\frac{\exp(\theta_{\mathbf{u}}^T \phi(\mathbf{u}))}{\mathcal{Z}(\theta_{\mathbf{u}})}, \quad (28)$$

where  $\mathcal{Z}(\theta_{\mathbf{u}}) = \int \exp(\theta_{\mathbf{u}}^T \phi(\mathbf{u})) d\mathbf{u}$  to ensure normalisation. We can then write the KL minimisation objective as follows,

$$\mathcal{F}_{\text{KL}} = \text{KL}(\tilde{p}(f) || q(f)) \quad (29)$$

$$= \int \tilde{p}(f) \log \frac{\tilde{p}(f)}{q(f)} df \quad (30)$$

$$= \langle \log \tilde{p}(f) \rangle_{\tilde{p}(f)} - \langle \log p(f_{\neq \mathbf{u}} | \mathbf{u}) \rangle_{\tilde{p}(f)} - \theta_{\mathbf{u}}^T \langle \phi(\mathbf{u}) \rangle_{\tilde{p}(f)} + \log \mathcal{Z}(\theta_{\mathbf{u}}). \quad (31)$$

Since  $p(f_{\neq \mathbf{u}} | \mathbf{u})$  is the prior conditional distribution, the only free parameter that controls our posterior approximation is  $\theta_{\mathbf{u}}$ . As such, to find  $\theta_{\mathbf{u}}$  that minimises  $\mathcal{F}_{\text{KL}}$ , we find the gradient of  $\mathcal{F}_{\text{KL}}$  w.r.t  $\theta_{\mathbf{u}}$  and set it to zero,

$$0 = \frac{d\mathcal{F}_{\text{KL}}}{d\theta_{\mathbf{u}}} = -\langle \phi(\mathbf{u}) \rangle_{\tilde{p}(f)} + \frac{d \log \mathcal{Z}(\theta_{\mathbf{u}})}{d\theta_{\mathbf{u}}} \quad (32)$$

$$= -\langle \phi(\mathbf{u}) \rangle_{\tilde{p}(f)} + \langle \phi(\mathbf{u}) \rangle_{q(u)}, \quad (33)$$

therefore,  $\langle \phi(\mathbf{u}) \rangle_{\tilde{p}(f)} = \langle \phi(\mathbf{u}) \rangle_{q(u)}$ . That is, though we are trying to perform the KL minimisation between two Gaussian processes, due to the special form of the posterior approximation, *it is sufficient to only match the moments at the inducing points  $\mathbf{u}$* .<sup>3</sup>

### Appendix C. Shortcuts to the moment matching equations

The most crucial step in Power-EP is the moment matching step as discussed above. This step can be done analytically for the Gaussian case, as the mean and covariance of the approximate posterior can be linked to the cavity distribution as follows,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}f}^{\setminus n} \frac{d \log \mathcal{Z}_{\text{tilted},n}}{d\mathbf{m}_f^{\setminus n}}, \quad (34)$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}f}^{\setminus n} \frac{d^2 \log \mathcal{Z}_{\text{tilted},n}}{d\mathbf{m}_f^{\setminus n,2}} \mathbf{V}_{f\mathbf{u}}^{\setminus n}, \quad (35)$$

where  $\mathcal{Z}_{\text{tilted},n}$  is the normaliser of the tilted distribution,

$$\mathcal{Z}_{\text{tilted},n} = \int q^{\setminus n}(f) p(y_n | f) df \quad (36)$$

$$= \int q^{\setminus n}(f) p(y_n | f_n) df \quad (37)$$

$$= \int q^{\setminus n}(f_n) p(y_n | f_n) df_n. \quad (38)$$

In words,  $\mathcal{Z}_{\text{tilted},n}$  only depends on the marginal distribution of the cavity process,  $q^{\setminus n}(f_n)$ , simplifying the moment matching equations above,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} \frac{d \log \mathcal{Z}_{\text{tilted},n}}{d\mathbf{m}_{f_n}^{\setminus n}}, \quad (39)$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} \frac{d^2 \log \mathcal{Z}_{\text{tilted},n}}{d\mathbf{m}_{f_n}^{\setminus n,2}} \mathbf{V}_{f_n \mathbf{u}}^{\setminus n}. \quad (40)$$

---

3. We can show that this condition gives the minimum of  $\mathcal{F}_{\text{KL}}$  by computing the second derivative.

We can rewrite the cross-covariance  $\mathbf{V}_{\mathbf{u}f_n}^{\setminus n} = \mathbf{V}_{\mathbf{u}}^{\setminus n} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}f_n}$ . We also note that,  $m_{f_n}^{\setminus n} = \mathbf{K}_{f_n\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{m}_{\mathbf{u}}^{\setminus n}$ , resulting in,

$$\frac{d \log \mathcal{Z}_{\text{tilted},n}}{d \mathbf{m}_{\mathbf{u}}^{\setminus n}} = \frac{d \log \mathcal{Z}_{\text{tilted},n}}{d m_{f_n}^{\setminus n}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}f_n}, \quad (41)$$

$$\frac{d \log \mathcal{Z}_{\text{tilted},n}}{d \mathbf{V}_{\mathbf{u}}^{\setminus n}} = \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}f_n} \frac{d^2 \log \mathcal{Z}_{\text{tilted},n}}{d m_{f_n}^{\setminus n,2}} \mathbf{K}_{f_n\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}. \quad (42)$$

Substituting these results back in eqs. 39 and 40, we obtain

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}}^{\setminus n} \frac{d \log \mathcal{Z}_{\text{tilted},n}}{d \mathbf{m}_{\mathbf{u}}^{\setminus n}}, \quad (43)$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}}^{\setminus n} \frac{d^2 \log \mathcal{Z}_{\text{tilted},n}}{d \mathbf{m}_{\mathbf{u}}^{\setminus n,2}} \mathbf{V}_{\mathbf{u}}^{\setminus n}. \quad (44)$$

Therefore, using eqs. 39 and 40, or eqs. 43 and 44 are equivalent in our approximation settings. In particular, we employ eqs. 39 and 40 for GP regression and classification, and use eqs. 43 and 44 for the GP state space model.

## Appendix D. Full derivation of the Power-EP procedure

We provide the full derivation of the Power-EP procedure in this section. We follow the derivation in (Qi et al., 2010) closely, but provide a clearer exposition and details how to get to each step used in the implementation, and how to handle powered/fractional deletion and update in Power-EP.

### D.1 Optimal factor parameterisation

We start by defining the approximate factors to be in natural parameter form as this makes it simple to combine and delete them,  $t_n(\mathbf{u}) = \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) = z_n \exp(\mathbf{u}^T \mathbf{T}_{1,n} - \frac{1}{2} \mathbf{u}^T \mathbf{T}_{2,n} \mathbf{u})$ . We initially consider full rank  $\mathbf{T}_{2,n}$ , but will show that the optimal form is rank 1.

The next goal is to relate these parameters to the approximate GP posterior. The approximate posterior over the pseudo-outputs has natural parameters  $\mathbf{T}_{1,\mathbf{u}} = \sum_n \mathbf{T}_{1,n}$  and  $\mathbf{T}_{2,\mathbf{u}} = \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} + \sum_n \mathbf{T}_{2,n}$ . This induces an approximate GP posterior with mean and covariance function,

$$m_{\mathbf{f}} = \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{T}_{2,\mathbf{u}}^{-1} \mathbf{T}_{1,\mathbf{u}} = \mathbf{K}_{\mathbf{f}\mathbf{u}} \gamma \quad (45)$$

$$V_{\mathbf{f}\mathbf{f}'} = \mathbf{K}_{\mathbf{f}\mathbf{f}'} - \mathbf{Q}_{\mathbf{f}\mathbf{f}'} + \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{T}_{2,\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{f}'} = \mathbf{K}_{\mathbf{f}\mathbf{f}'} - \mathbf{K}_{\mathbf{f}\mathbf{u}} \beta \mathbf{K}_{\mathbf{u}\mathbf{f}'}. \quad (46)$$

where  $\gamma$  and  $\beta$  are likelihood-dependent terms we wish to store and update using PEP;  $\gamma$  and  $\beta$  fully specify the approximate posterior.

**Deletion step:** The cavity for data point  $n$ ,  $q^{\setminus n}(f) \propto q^*(f)/t_n^\alpha(\mathbf{u})$ , has a similar form to the posterior, but the natural parameters are modified by the deletion,  $\mathbf{T}_{1,\mathbf{u}}^{\setminus n} = \mathbf{T}_{1,\mathbf{u}} - \alpha \mathbf{T}_{1,n}$

and  $\mathbf{T}_{2,\mathbf{u}}^{\setminus n} = \mathbf{T}_{2,\mathbf{u}} - \alpha \mathbf{T}_{2,n}$ , yielding a new mean and covariance function

$$m_{\mathbf{f}}^{\setminus n} = \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{T}_{2,\mathbf{u}}^{\setminus n, -1} \mathbf{T}_{1,\mathbf{u}}^{\setminus n} = \mathbf{K}_{\mathbf{f}\mathbf{u}} \gamma^{\setminus n} \quad (47)$$

$$V_{\mathbf{f}\mathbf{f}'}^{\setminus n} = \mathbf{K}_{\mathbf{f}\mathbf{f}'} - \mathbf{Q}_{\mathbf{f}\mathbf{f}'} + \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{T}^{\setminus n, -1} \mathbf{T}_{2,\mathbf{u}}^{\setminus n, -1} \mathbf{K}_{\mathbf{u}\mathbf{f}'} = \mathbf{K}_{\mathbf{f}\mathbf{f}'} - \mathbf{K}_{\mathbf{f}\mathbf{u}} \beta^{\setminus n} \mathbf{K}_{\mathbf{u}\mathbf{f}'}. \quad (48)$$

**Projection step:** The central step in Power EP is the projection step. Obtaining the new approximate unnormalised posterior  $q^*(f)$  such that  $\text{KL}(\tilde{p}(f) || q^*(f))$  is minimised would naïvely appear intractable. Fortunately, as shown in the previous section, because of the structure of the approximate posterior,  $q(f) = p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})$ , the objective,  $\text{KL}(\tilde{p}(f) || q^*(f))$  is minimised when  $\mathbb{E}_{\tilde{p}(f)}[\phi(\mathbf{u})] = \mathbb{E}_{q(\mathbf{u})}[\phi(\mathbf{u})]$ , where  $\phi(\mathbf{u})$  are the sufficient statistics, that is when the moments at the pseudo-inputs are matched. This is the central result from which computational savings are derived. Furthermore, this moment matching condition would appear to necessitate computation of a set of integrals to find the zeroth, first and second moments. Using results from the previous section simplifies and provides the following shortcuts,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n} \frac{d \log \tilde{Z}_n}{dm_{\mathbf{f}_n}^{\setminus n}} \quad (49)$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n} \frac{d^2 \log \tilde{Z}_n}{d(m_{\mathbf{f}_n}^{\setminus n})^2} \mathbf{V}_{\mathbf{f}_n \mathbf{u}}^{\setminus n}. \quad (50)$$

where  $\log \tilde{Z}_n = \log \mathbb{E}_{q^{\setminus n}(f)}[p^\alpha(y_n | \mathbf{f}_n)]$  is the log-normaliser of the tilted distribution.

**Update step:** Having computed the new approximate posterior, the fractional approximate factor  $t_{n,\text{new}}(\mathbf{u}) = q^*(f)/q^{\setminus n}(f)$  can be straightforwardly obtained, resulting in,

$$\mathbf{T}_{1,n,\text{new}} = \mathbf{V}_{\mathbf{u}}^{-1} \mathbf{m}_{\mathbf{u}} - \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \mathbf{m}_{\mathbf{u}}^{\setminus n} \quad (51)$$

$$\mathbf{T}_{2,n,\text{new}} = \mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \quad (52)$$

$$z_n^\alpha = \tilde{Z}_n \exp(\mathcal{G}_{q^{\setminus n}}(\mathbf{u}) - \mathcal{G}_{q^*(\mathbf{u})}), \quad (53)$$

where  $\mathcal{G}_{\tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)} = \int \tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2) d\mathbf{u}$ . Let  $d_1 = \frac{d \log \tilde{Z}_n}{dm_{\mathbf{f}_n}^{\setminus n}}$  and  $d_2 = \frac{d^2 \log \tilde{Z}_n}{d(m_{\mathbf{f}_n}^{\setminus n})^2}$ . Using eq. (21) and eq. (50), we have,

$$\mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} = -\mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n} \left[ d_2^{-1} + \mathbf{V}_{\mathbf{f}_n \mathbf{u}}^{\setminus n} \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n} \right]^{-1} \mathbf{V}_{\mathbf{f}_n \mathbf{u}}^{\setminus n} \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \quad (54)$$

Let  $v_n = \alpha(-d_2^{-1} - \mathbf{V}_{\mathbf{f}_n \mathbf{u}}^{\setminus n} \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n})$ , and  $\mathbf{w}_n = \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n}$ . Combining eq. (54) and eq. (52) gives

$$\mathbf{T}_{2,n,\text{new}} = \mathbf{w}_n \alpha v_n^{-1} \mathbf{w}_n^\top \quad (55)$$

At convergence, we have  $t_n(\mathbf{u})^\alpha = t_{n,\text{new}}(\mathbf{u})$ , hence  $\mathbf{T}_{2,n} = \mathbf{w}_n v_n^{-1} \mathbf{w}_n^\top$ . In words,  $\mathbf{T}_{2,n}$  is optimally a rank-1 matrix. Note that,

$$\mathbf{w}_n = \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \mathbf{V}_{\mathbf{u}\mathbf{f}_n}^{\setminus n} \quad (56)$$

$$= (\mathbf{K}_{\mathbf{u}\mathbf{u}} - \mathbf{K}_{\mathbf{u}\mathbf{u}} \beta^{\setminus n} \mathbf{K}_{\mathbf{u}\mathbf{u}})^{-1} (\mathbf{K}_{\mathbf{u}\mathbf{f}_n} - \mathbf{K}_{\mathbf{u}\mathbf{u}} \beta^{\setminus n} \mathbf{K}_{\mathbf{u}\mathbf{f}_n}) \quad (57)$$

$$= \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} (\mathbf{I} - \mathbf{K}_{\mathbf{u}\mathbf{u}} \beta^{\setminus n})^{-1} (\mathbf{I} - \mathbf{K}_{\mathbf{u}\mathbf{u}} \beta^{\setminus n}) \mathbf{K}_{\mathbf{u}\mathbf{f}_n} \quad (58)$$

$$= \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{f}_n}. \quad (59)$$

Using eq. (49) and eq. (55) gives,

$$\mathbf{V}_u^{-1} \mathbf{m}_u = (\mathbf{V}_u^{\setminus n, -1} + \mathbf{w}_n \alpha v_n^{-1} \mathbf{w}_n^\top) (\mathbf{m}_u^{\setminus n} + \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_1) \quad (60)$$

$$= \mathbf{V}_u^{\setminus n, -1} \mathbf{m}_u^{\setminus n} + \mathbf{w}_n \alpha v_n^{-1} \mathbf{w}_n^\top \mathbf{m}_u^{\setminus n} + \mathbf{V}_u^{\setminus n, -1} \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_1 + \mathbf{w}_n \alpha v_n^{-1} \mathbf{w}_n^\top \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_1 \quad (61)$$

Substituting this result into eq. (51),

$$\mathbf{T}_{1,n,\text{new}} = \mathbf{V}_u^{-1} \mathbf{m}_u - \mathbf{V}_u^{\setminus n, -1} \mathbf{m}_u^{\setminus n} \quad (62)$$

$$= \mathbf{w}_n \alpha v_n^{-1} \mathbf{w}_n^\top \mathbf{m}_u^{\setminus n} + \mathbf{V}_u^{\setminus n, -1} \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_1 + \mathbf{w}_n \alpha v_n^{-1} \mathbf{w}_n^\top \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_1 \quad (63)$$

$$= \mathbf{w}_n \alpha v_n^{-1} \left( \mathbf{w}_n^\top \mathbf{m}_u^{\setminus n} + d_1 v_n / \alpha + \mathbf{w}_n^\top \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_1 \right). \quad (64)$$

Let  $\mathbf{T}_{1,n,\text{new}} = \mathbf{w}_n \alpha v_n^{-1} g_n$ , we obtain,

$$g_n = -\frac{d_1}{d_2} + \mathbf{K}_{f_n \mathbf{u}} \gamma^{\setminus n}. \quad (65)$$

At convergence,  $\mathbf{T}_{1,n} = \mathbf{w}_n v_n^{-1} g_n$ . Re-writing the form of the approximate factor using  $\mathbf{T}_{1,n}$  and  $\mathbf{T}_{2,n}$  at convergence,

$$t_n(\mathbf{u}) = \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) \quad (66)$$

$$= z_n \exp(\mathbf{u}^\top \mathbf{T}_{1,n} - \frac{1}{2} \mathbf{u}^\top \mathbf{T}_{2,n} \mathbf{u}) \quad (67)$$

$$= z_n \exp(\mathbf{u}^\top \mathbf{w}_n v_n^{-1} g_n - \frac{1}{2} \mathbf{u}^\top \mathbf{w}_n v_n^{-1} \mathbf{w}_n^\top \mathbf{u}) \quad (68)$$

As a result, the minimal and simplest way to parameterise the approximate factor is  $t_n(\mathbf{u}) = \tilde{z}_n \mathcal{N}(\mathbf{w}_n^\top \mathbf{u}; g_n, v_n) = \tilde{z}_n \mathcal{N}(\mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{u}; g_n, v_n)$ , where  $g_n$  and  $v_n$  are scalars, resulting in a significant memory saving compared to the parameterisation using  $\mathbf{T}_{1,n}$  and  $\mathbf{T}_{2,n}$ .

## D.2 Projection

We now recall the update equations in the projection step (eqns. 49 and 50):

$$\mathbf{m}_u = \mathbf{m}_u^{\setminus n} + \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_1, \quad (69)$$

$$\mathbf{V}_u = \mathbf{V}_u^{\setminus n} + \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_2 \mathbf{V}_{f_n \mathbf{u}}^{\setminus n}. \quad (70)$$

Note that:

$$\mathbf{m}_u = \mathbf{K}_{\mathbf{u}\mathbf{u}} \gamma, \quad (71)$$

$$\mathbf{V}_u = \mathbf{K}_{\mathbf{u}\mathbf{u}} - \mathbf{K}_{\mathbf{u}\mathbf{u}} \beta \mathbf{K}_{\mathbf{u}\mathbf{u}}, \quad (72)$$

and

$$\mathbf{m}_u^{\setminus n} = \mathbf{K}_{\mathbf{u}\mathbf{u}} \gamma^{\setminus n}, \quad (73)$$

$$\mathbf{V}_u^{\setminus n} = \mathbf{K}_{\mathbf{u}\mathbf{u}} - \mathbf{K}_{\mathbf{u}\mathbf{u}} \beta^{\setminus n} \mathbf{K}_{\mathbf{u}\mathbf{u}}. \quad (74)$$

Using these results, we can convert the update for the mean and covariance,  $\mathbf{m}_{\mathbf{u}}$  and  $\mathbf{V}_{\mathbf{u}}$ , into an update for  $\gamma$  and  $\beta$ ,

$$\gamma = \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{m}_{\mathbf{u}} \quad (75)$$

$$= \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{m}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_1) \quad (76)$$

$$= \gamma^{\setminus n} + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_1, \text{ and} \quad (77)$$

$$\beta = \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}}) \mathbf{K}_{\mathbf{uu}}^{-1} \quad (78)$$

$$= \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}}^{\setminus n} - \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_2 \mathbf{V}_{\mathbf{f}_n \mathbf{u}}^{\setminus n}) \mathbf{K}_{\mathbf{uu}}^{-1} \quad (79)$$

$$= \beta^{\setminus n} - \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{uf}_n}^{\setminus n} d_2 \mathbf{V}_{\mathbf{f}_n \mathbf{u}}^{\setminus n} \mathbf{K}_{\mathbf{uu}}^{-1} \quad (80)$$

### D.3 Deletion step

Finally, we present how deletion might be accomplished. One direct approach to this step is to divide out the cavity from the cavity, that is,

$$q^{\setminus n}(f) \propto \frac{q(f)}{t_n^\alpha(\mathbf{u})} = \frac{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})}{t_n^\alpha(\mathbf{u})} = p(f_{\neq \mathbf{u}} | \mathbf{u}) q^{\setminus n}(\mathbf{u}). \quad (81)$$

Instead, we use an alternative using the KL minimisation as used in (Qi et al., 2010), by realising that doing this will result in an identical outcome as the direct approach since the factor and distributions are Gaussian. Furthermore, we can re-use results from the projection and inclusion steps, by simply swapping the quantities and negating the site approximation variance. In particular, we present projection and deletion side-by-side, to facilitate the comparison,

$$\text{Projection: } q(f) \approx q^{\setminus n}(f) p(y_n | \mathbf{f}_n) \quad (82)$$

$$\text{Deletion: } q^{\setminus n}(f) \propto q(f) \frac{1}{t_n^\alpha(\mathbf{u})} \quad (83)$$

The projection step minimises the KL between the LHS and RHS while moment matching, to get  $q(f)$ . We would like to do the same for the deletion step to find  $q^{\setminus n}(f)$ , and thus reuse the same moment matching results for  $\gamma$  and  $\beta$  with some modifications.

Our task will be to reuse Equations 77 and 80, the moment matching equations in  $\gamma$  and  $\beta$ . We have two differences to account for. Firstly, we need to change any uses of the parameters of the cavity distribution to the parameters of the approximate posterior,  $\mathbf{V}_{\mathbf{uf}_n}^{\setminus n}$  to  $\mathbf{V}_{\mathbf{uf}_n}$ ,  $\gamma^{\setminus n}$  to  $\gamma$  and  $\beta^{\setminus n}$  to  $\beta$ . This is the equivalent of re-deriving the entire projection operation, while swapping the symbols (and quantities) for the cavity and the full distribution. Secondly, the derivatives  $d_1$  and  $d_2$  are different here, as

$$\log \tilde{Z}_n = \log \int q(f) \frac{1}{t_n^\alpha(\mathbf{u})} df \quad (84)$$

Now, we note

$$\frac{1}{t_n(\mathbf{u})} \propto \frac{1}{\mathcal{N}^\alpha(\mathbf{w}_n^\top \mathbf{u}; g_n, v_n)} \quad (85)$$

$$\propto \frac{1}{\exp\left(-\frac{\alpha}{2}v_n^{-1}(\mathbf{w}_n^\top \mathbf{u} - g_n)^2\right)} \quad (86)$$

$$= \exp\left(\frac{1}{2}\alpha v_n^{-1}(\mathbf{w}_n^\top \mathbf{u} - g_n)^2\right) \quad (87)$$

$$\propto \mathcal{N}(\mathbf{w}_n^\top \mathbf{u}; g_n, -v_n/\alpha) \quad (88)$$

Then we obtain the derivatives of  $\log \tilde{Z}_n$

$$\tilde{d}_2 = \frac{d^2 \log \tilde{Z}_n}{dm_{f_n}^2} = -[\mathbf{K}_{f_n, \mathbf{u}} \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}, f_n} - \mathbf{K}_{f_n, \mathbf{u}} \beta \mathbf{K}_{\mathbf{u}, f_n} - v_n/\alpha]^{-1} \quad (89)$$

$$\tilde{d}_1 = \frac{d \log \tilde{Z}_n}{dm_{f_n}} = (\mathbf{K}_{f_n, \mathbf{u}} \gamma - g_n) \tilde{d}_2 \quad (90)$$

Putting the above results together, we obtain,

$$\gamma^{\setminus n} = \gamma + \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{V}_{\mathbf{u}f_n} \tilde{d}_1, \text{ and} \quad (91)$$

$$\beta^{\setminus n} = \beta - \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{V}_{\mathbf{u}f_n} \tilde{d}_2 \mathbf{V}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \quad (92)$$

#### D.4 Summary of the PEP procedure

We summarise here the key steps and equations that we have obtained, that are used in the implementation:

1. Initialise the parameters:  $\{g_n = 0\}_{n=1}^N$ ,  $\{v_n = \infty\}_{n=1}^N$ ,  $\gamma = \mathbf{0}_{M \times 1}$  and  $\beta = \mathbf{0}_{M \times M}$
2. Loop through all data points until convergence:

- (a) Deletion step: find  $\gamma^{\setminus n}$  and  $\beta^{\setminus n}$

$$\gamma^{\setminus n} = \gamma + \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{V}_{\mathbf{u}f_n} \tilde{d}_1, \text{ and} \quad (93)$$

$$\beta^{\setminus n} = \beta - \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{V}_{\mathbf{u}f_n} \tilde{d}_2 \mathbf{V}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \quad (94)$$

- (b) Projection step: find  $\gamma$  and  $\beta$

$$\gamma = \gamma^{\setminus n} + \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_1, \quad (95)$$

$$\beta = \beta^{\setminus n} - \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} d_2 \mathbf{V}_{f_n \mathbf{u}}^{\setminus n} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \quad (96)$$

- (c) Update step: find  $g_{n,\text{new}}$  and  $v_{n,\text{new}}$

$$g_{n,\text{new}} = -\frac{d_1}{d_2} + \mathbf{K}_{f_n \mathbf{u}} \gamma^{\setminus n}, \quad (97)$$

$$v_{n,\text{new}} = -d_2^{-1} - \mathbf{V}_{f_n \mathbf{u}}^{\setminus n} \mathbf{V}_{\mathbf{u}}^{\setminus n, -1} \mathbf{V}_{\mathbf{u}f_n}^{\setminus n} \quad (98)$$



and parameters for the full factor,

$$v_n \leftarrow (v_{n,\text{new}}^{-1} + (1 - \alpha)v_n^{-1})^{-1} \quad (99)$$

$$g_n \leftarrow v_n(g_{n,\text{new}}v_{n,\text{new}}^{-1} + (1 - \alpha)g_nv_n^{-1}) \quad (100)$$

## Appendix E. Power-EP energy for sparse GP regression and classification

The Power-EP procedure gives an approximate marginal likelihood, which is the negative Power-EP energy, as follows,

$$\mathcal{F} = \phi_{\text{post}} - \phi_{\text{prior}} + \frac{1}{\alpha} \sum_n \log \mathcal{Z}_{\text{tilted},n} + \phi_{\text{cav},n} - \phi_{\text{post}} \quad (101)$$

where  $\phi_{\text{post}}$  is the log-normaliser of the approximate posterior, that is,

$$\phi_{\text{post}} = \log \int p(f_{\neq \mathbf{u}} | \mathbf{u}) \exp(\theta_{\mathbf{u}}^T \phi(\mathbf{u})) df_{\neq \mathbf{u}} d\mathbf{u} \quad (102)$$

$$= \log \int \exp(\theta_{\mathbf{u}}^T \phi(\mathbf{u})) d\mathbf{u} \quad (103)$$

$$= \frac{M}{2} \log(2\pi) + \frac{1}{2} \log |\mathbf{V}| + \frac{1}{2} \mathbf{m}^T \mathbf{V}^{-1} \mathbf{m}, \quad (104)$$

where  $\mathbf{m}$  and  $\mathbf{V}$  are the mean and covariance of the posterior distribution over  $\mathbf{u}$ , respectively. Similarly,

$$\phi_{\text{cav},n} = \frac{M}{2} \log(2\pi) + \frac{1}{2} \log |\mathbf{V}_{\text{cav},n}| + \frac{1}{2} \mathbf{m}_{\text{cav},n}^T \mathbf{V}_{\text{cav},n}^{-1} \mathbf{m}_{\text{cav},n}, \quad (105)$$

$$\text{and } \phi_{\text{prior}} = \frac{M}{2} \log(2\pi) + \frac{1}{2} \log |\mathbf{K}_{\mathbf{uu}}|. \quad (106)$$

Finally,  $\log \mathcal{Z}_{\text{tilted},n}$  is the log-normalising constant of the tilted distribution,

$$\log \mathcal{Z}_{\text{tilted}} = \log \int q_{\text{cav}}(f) p^\alpha(y_n | f) df \quad (107)$$

$$= \log \int p(f_{\neq \mathbf{u}} | \mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^\alpha(y_n | f) df_{\neq \mathbf{u}} d\mathbf{u} \quad (108)$$

$$= \log \int p(f_n | \mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^\alpha(y_n | f_n) df_n d\mathbf{u} \quad (109)$$

Next, we can write down the form of the natural parameters of the approximate posterior and the cavity distribution, based on the approximate factor's parameters, as follows,

$$\mathbf{V}^{-1} = \mathbf{K}_{\mathbf{uu}}^{-1} + \sum_i \mathbf{w}_i \tau_i \mathbf{w}_i^T \quad (110)$$

$$\mathbf{V}^{-1} \mathbf{m} = \sum_i \mathbf{w}_i \tau_i \tilde{y}_i \quad (111)$$

$$\mathbf{V}_{\text{cav},n}^{-1} = \mathbf{V}^{-1} - \alpha \mathbf{w}_n \tau_n \mathbf{w}_n^T \quad (112)$$

$$\mathbf{V}_{\text{cav},n}^{-1} \mathbf{m}_{\text{cav},n} = \mathbf{V}^{-1} \mathbf{m} - \alpha \mathbf{w}_n \tau_n g_n \quad (113)$$

Note that  $\tau_i := v_i^{-1}$ . Using eq. (23) and eq. (112) gives,

$$\mathbf{V}_{\text{cav},n} = \mathbf{V} + \frac{\mathbf{V}\mathbf{w}_n\alpha\tau_n\mathbf{w}_n^\top\mathbf{V}}{1 - \mathbf{w}_n^\top\alpha\tau_n\mathbf{V}\mathbf{w}_n}. \quad (114)$$

Using eq. (24) and eq. (112) gives,

$$\log \det(\mathbf{V}_{\text{cav},n}) = \log \det(\mathbf{V}) - \log(1 - \mathbf{w}_n^\top\alpha\tau_n\mathbf{V}\mathbf{w}_n). \quad (115)$$

Substituting eq. (114) and eq. (115) back to eq. (105) results in,

$$\begin{aligned} \phi_{\text{cav},n} = & \frac{M}{2} \log(2\pi) + \frac{1}{2} \log \det(\mathbf{V}) + \frac{1}{2} \mathbf{m}^\top \mathbf{V}^{-1} \mathbf{m} \\ & - \frac{1}{2} \log(1 - \mathbf{w}_n^\top\alpha\tau_n\mathbf{V}\mathbf{w}_n) + \frac{1}{2} \frac{\mathbf{m}^\top \mathbf{w}_n\alpha\tau_n\mathbf{w}_n^\top \mathbf{m}}{1 - \mathbf{w}_n^\top\alpha\tau_n\mathbf{V}\mathbf{w}_n} \\ & + \frac{1}{2} g_n\alpha\tau_n\mathbf{w}_n^\top\mathbf{V}_{\text{cav},n}\mathbf{w}_n\alpha\tau_n g_n - g_n\alpha\tau_n\mathbf{w}_n^\top\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m} \end{aligned} \quad (116)$$

We now plug the above result back into the approximate marginal likelihood, yielding,

$$\begin{aligned} \mathcal{F} = & \frac{1}{2} \log |\mathbf{V}| + \frac{1}{2} \mathbf{m}^\top \mathbf{V}^{-1} \mathbf{m} - \frac{1}{2} \log |\mathbf{K}_{\text{uu}}| + \frac{1}{\alpha} \sum_n \log \mathcal{Z}_{\text{tilted},n} \\ & + \sum_n \left[ -\frac{1}{2\alpha} \log(1 - \mathbf{w}_n^\top\alpha\tau_n\mathbf{V}\mathbf{w}_n) + \frac{1}{2} \frac{\mathbf{m}^\top \mathbf{w}_n\tau_n\mathbf{w}_n^\top \mathbf{m}}{1 - \mathbf{w}_n^\top\alpha\tau_n\mathbf{V}\mathbf{w}_n} \right. \\ & \left. + \frac{1}{2} g_n\tau_n\mathbf{w}_n^\top\mathbf{V}_{\text{cav},n}\mathbf{w}_n\alpha\tau_n g_n - g_n\tau_n\mathbf{w}_n^\top\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m} \right] \end{aligned} \quad (117)$$

### E.1 Regression

We have shown in the previous section that the fixed point solution of the Power-EP iterations can be obtained analytically for the regression case,  $g_n = y_n$  and  $\tau_n^{-1} = d_n = \alpha(K_{f_n f_n} - \mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\text{uu}}^{-1} \mathbf{K}_{\mathbf{u} f_n}) + \sigma_y^2$ . Crucially, we can obtain a closed form expression for  $\log \mathcal{Z}_{\text{tilted},n}$ ,

$$\log \mathcal{Z}_{\text{tilted},n} = -\frac{\alpha}{2} \log(2\pi\sigma_y^2) + \frac{1}{2} \log(\sigma_y^2) - \frac{1}{2} \log(\alpha v_n + \sigma_y^2) - \frac{1}{2} \frac{(y_n - \mu_n)^2}{v_n + \sigma_y^2/\alpha} \quad (118)$$

where  $\mu_n = \mathbf{w}_n^\top \mathbf{m}_{\text{cav}} = \mathbf{w}_n^\top \mathbf{V}_{\text{cav}} (\mathbf{V}^{-1} \mathbf{m} - \mathbf{w}_n \alpha \tau_n y_n)$  and  $v_n = \frac{d_n - \sigma_y^2}{\alpha} + \mathbf{w}_n^\top \mathbf{V}_{\text{cav}} \mathbf{w}_n$ . We can therefore simplify the approximate marginal likelihood  $\mathcal{F}$  further,

$$\begin{aligned} \mathcal{F} = & \frac{1}{2} \log |\mathbf{V}| + \frac{1}{2} \mathbf{m}^\top \mathbf{V}^{-1} \mathbf{m} - \frac{1}{2} \log |\mathbf{K}_{\text{uu}}| + \sum_n \left[ -\frac{1}{2} \log(2\pi\sigma_y^2) + \frac{1}{2\alpha} \log \sigma_y^2 - \frac{1}{2\alpha} \log d_n - \frac{y_n^2}{2d_n} \right] \\ = & -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{D} + \mathbf{Q}_{\text{ff}}| - \frac{1}{2} \mathbf{y}^\top (\mathbf{D} + \mathbf{Q}_{\text{ff}})^{-1} \mathbf{y} - \frac{1-\alpha}{2\alpha} \sum_n \log\left(\frac{d_n}{\sigma_y^2}\right), \end{aligned} \quad (119)$$

where  $\mathbf{Q}_{\text{ff}} = \mathbf{K}_{\text{fu}} \mathbf{K}_{\text{uu}}^{-1} \mathbf{K}_{\text{uf}}$  and  $\mathbf{D}$  is a diagonal matrix,  $\mathbf{D}_{nn} = d_n$ .

When  $\alpha = 1$ , the approximate marginal likelihood takes the same form as the FITC marginal likelihood,

$$\mathcal{F} = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{D} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2} \mathbf{y}^T (\mathbf{D} + \mathbf{Q}_{\mathbf{ff}})^{-1} \mathbf{y} \quad (120)$$

where  $\mathbf{D}_{nn} = d_n = K_{f_n f_n} - \mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{u} f_n} + \sigma_y^2$ .

When  $\alpha$  tends to 0, we have,

$$\lim_{\alpha \rightarrow 0} \frac{1-\alpha}{2\alpha} \sum_n \log\left(\frac{d_n}{\sigma_y^2}\right) = \frac{1}{2} \sum_n \lim_{\alpha \rightarrow 0} \frac{\log(1 + \alpha \frac{g_n}{\sigma_y^2})}{\alpha} = \frac{\sum_n h_n}{2\sigma_y^2}, \quad (121)$$

where  $h_n = K_{f_n f_n} - \mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{u} f_n}$ . Therefore,

$$\mathcal{F} = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\sigma_y^2 \mathbf{I} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2} \mathbf{y}^T (\sigma_y^2 \mathbf{I} + \mathbf{Q}_{\mathbf{ff}})^{-1} \mathbf{y} - \frac{\sum_n h_n}{2\sigma_y^2}, \quad (122)$$

which is the variational lower bound of Titsias (Titsias, 2009b).

## E.2 Classification

In contrast to the regression case, the approximate marginal likelihood for classification cannot be simplified due to the non-Gaussian likelihood. Specifically,  $\log \mathcal{Z}_{\text{tilted},n}$  is not analytically tractable, except when  $\alpha = 1$  and the classification link function is the Gaussian CDF. However, this quantity can be evaluated numerically, using sampling or Gauss-Hermite quadrature, since it only involves a one-dimensional integral.

We now consider the case when  $\alpha$  tends to 0 and verify that in such case the approximate marginal likelihood becomes the variational lower bound. We first find the limits of individual terms in eq. (117):

$$\lim_{\alpha \rightarrow 0} -\frac{1}{2\alpha} \log(1 - \mathbf{w}_n^T \alpha \tau_n \mathbf{V} \mathbf{w}_n) = \frac{1}{2} \mathbf{w}_n^T \tau_n \mathbf{V} \mathbf{w}_n \quad (123)$$

$$\left. \frac{1}{2} \frac{\mathbf{m}^T \mathbf{w}_n \tau_n \mathbf{w}_n^T \mathbf{m}}{1 - \mathbf{w}_n^T \alpha \tau_n \mathbf{V} \mathbf{w}_n} \right|_{\alpha=0} = \frac{1}{2} \mathbf{m}^T \mathbf{w}_n \tau_n \mathbf{w}_n^T \mathbf{m} \quad (124)$$

$$\left. \frac{1}{2} g_n \tau_n \mathbf{w}_n^T \mathbf{V}_{\text{cav},n} \mathbf{w}_n \alpha \tau_n g_n \right|_{\alpha=0} = 0 \quad (125)$$

$$\left. -g_n \tau_n \mathbf{w}_n^T \mathbf{V}_{\text{cav},n} \mathbf{V}^{-1} \mathbf{m} \right|_{\alpha=0} = -g_n \tau_n \mathbf{w}_n^T \mathbf{m}. \quad (126)$$

We turn our attention to  $\log \mathcal{Z}_{\text{tilted},n}$ . First, we expand  $p^\alpha(y_n|f_n)$  using eq. (26):

$$p^\alpha(y_n|f_n) = \exp(\alpha \log p(y_n|f_n)) \quad (127)$$

$$= 1 + \alpha \log p(y_n|f_n) + \xi(\alpha^2). \quad (128)$$

Substituting this result back into  $\log \mathcal{Z}_{\text{tilted}}/\alpha$  gives,

$$\frac{1}{\alpha} \log \mathcal{Z}_{\text{tilted}} = \frac{1}{\alpha} \log \int p(f_n|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^\alpha(y_n|f_n) df_n d\mathbf{u} \quad (129)$$

$$= \frac{1}{\alpha} \log \int p(f_n|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) [1 + \alpha \log p(y_n|f_n) + \xi(\alpha^2)] df_n d\mathbf{u} \quad (130)$$

$$= \frac{1}{\alpha} \log \left[ 1 + \alpha \int p(f_n|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) \log p(y_n|f_n) df_n d\mathbf{u} + \alpha^2 \xi(1) \right] \quad (131)$$

$$= \frac{1}{\alpha} \left[ \alpha \int p(f_n|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) \log p(y_n|f_n) df_n d\mathbf{u} + \alpha^2 \xi(1) \right] \quad (132)$$

$$= \int p(f_n|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) \log p(y_n|f_n) df_n d\mathbf{u} + \alpha \xi(1). \quad (133)$$

Therefore,

$$\lim_{\alpha \rightarrow 0} \frac{1}{\alpha} \log \mathcal{Z}_{\text{tilted}} = \int p(f_n|\mathbf{u}) q(\mathbf{u}) \log p(y_n|f_n) df_n d\mathbf{u}. \quad (134)$$

Putting these results into eq. (117), we obtain,

$$\begin{aligned} \mathcal{F} &= \frac{1}{2} \log |\mathbf{V}| + \frac{1}{2} \mathbf{m}^\top \mathbf{V}^{-1} \mathbf{m} - \frac{1}{2} \log |\mathbf{K}_{\mathbf{uu}}| \\ &\quad + \sum_n \frac{1}{2} \mathbf{w}_n^\top \tau_n \mathbf{V} \mathbf{w}_n + \frac{1}{2} \mathbf{m}^\top \mathbf{w}_n \tau_n \mathbf{w}_n^\top \mathbf{m} - g_n \tau_n \mathbf{w}_n^\top \mathbf{m} + \int p(f_n|\mathbf{u}) q(\mathbf{u}) \log p(y_n|f_n) df_n d\mathbf{u} \\ &= \frac{1}{2} \log |\mathbf{V}| + \frac{1}{2} \mathbf{m}^\top \mathbf{V}^{-1} \mathbf{m} - \frac{1}{2} \log |\mathbf{K}_{\mathbf{uu}}| + \frac{1}{2} \mathbf{m}^\top (\mathbf{V}^{-1} - \mathbf{K}_{\mathbf{uu}}^{-1}) \mathbf{m} - \mathbf{m}^\top \mathbf{V}^{-1} \mathbf{m} \\ &\quad + \sum_n \frac{1}{2} \mathbf{w}_n^\top \tau_n \mathbf{V} \mathbf{w}_n + \int p(f_n|\mathbf{u}) q(\mathbf{u}) \log p(y_n|f_n) df_n d\mathbf{u} \\ &= \frac{1}{2} \log |\mathbf{V}| - \frac{1}{2} \mathbf{m}^\top \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{m} - \frac{1}{2} \log |\mathbf{K}_{\mathbf{uu}}| + \sum_n \frac{1}{2} \mathbf{w}_n^\top \tau_n \mathbf{V} \mathbf{w}_n + \sum_n \int p(f_n|\mathbf{u}) q(\mathbf{u}) \log p(y_n|f_n) df_n d\mathbf{u}. \end{aligned} \quad (135)$$

We now write down the evidence lower bound of the global variational approach of Titsias (Titsias, 2009b), as applied to the classification case (Hensman et al., 2015),

$$\mathcal{F}_{\text{VFE}} = -\text{KL}(q(\mathbf{u})||p(\mathbf{u})) + \sum_n \int p(f_n|\mathbf{u}) q(\mathbf{u}) \log p(y_n|f_n) df_n d\mathbf{u} \quad (136)$$

where

$$\begin{aligned} -\text{KL}(q(\mathbf{u})||p(\mathbf{u})) &= -\frac{1}{2} \text{trace}(\mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}) - \frac{1}{2} \mathbf{m}^\top \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{m} + \frac{M}{2} - \frac{1}{2} \log |\mathbf{K}_{\mathbf{uu}}| + \frac{1}{2} \log |\mathbf{V}| \\ &= -\frac{1}{2} \text{trace}([\mathbf{V}^{-1} - \sum_n \mathbf{w}_n \tau_n \mathbf{w}_n^\top] \mathbf{V}) - \frac{1}{2} \mathbf{m}^\top \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{m} + \frac{M}{2} - \frac{1}{2} \log |\mathbf{K}_{\mathbf{uu}}| + \frac{1}{2} \log |\mathbf{V}| \\ &= \frac{1}{2} \text{trace}(\sum_n \mathbf{w}_n \tau_n \mathbf{w}_n^\top \mathbf{V}) - \frac{1}{2} \mathbf{m}^\top \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{m} - \frac{1}{2} \log |\mathbf{K}_{\mathbf{uu}}| + \frac{1}{2} \log |\mathbf{V}|. \end{aligned} \quad (137)$$

Therefore,  $\mathcal{F}_{\text{VFE}}$  is **identical** to the limit of the approximate marginal likelihood provided by power-EP as shown in eq. (135).

## Appendix F. Power-EP for the Gaussian process state space model

We recap here the description of the model and include the full details of the inference procedure. The GPSSM can be compactly represented as follows in the case where the dynamical noise is assumed Gaussian,

$$\begin{aligned} p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; f(\mathbf{x}_{t-1}), \sigma_x^2), \\ p(\mathbf{y}_t|\mathbf{x}_t) &= \mathcal{N}(\mathbf{y}_t; \mathbf{C}\mathbf{x}_t, \mathbf{R}_y), \end{aligned}$$

where  $\mathbf{x}$  and  $\mathbf{y}$  are the latent variables and the measurements respectively. The exact and intractable posterior over the latent function  $f$  and the hidden states  $\mathbf{x}$  is as follows,

$$p(\mathbf{x}, f) \propto p(\mathbf{x}_0)p(f) \prod_t p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) \prod_t p(\mathbf{y}_t|\mathbf{x}_t)$$

We posit the following approximate posterior, which allows us to employ Power-EP to perform inference,

$$q(\mathbf{x}, f) \propto p(\mathbf{x}_0)p(f_{\neq \mathbf{u}}|\mathbf{u}) \prod_t \lambda_t(\mathbf{x}_{t-1})\beta_t(\mathbf{x}_t)\gamma_t(\mathbf{u}) \prod_t p(\mathbf{y}_t|\mathbf{x}_t),$$

where  $\lambda_t, \beta_t$  and  $\gamma_t$  are approximate factors, and  $\mathbf{u}$  are the *inducing points*. The inducing points  $\mathbf{u}$  are used in this context to facilitate analytically tractable message passing, and, if  $|\mathbf{u}| < T$ , to sidestep the cubic cost of the GP. Note that we did not have separate approximate factors for the emission factors, since the correct factors are assumed Gaussian, the approximation becomes exact here.

### F.1 The Power-EP procedure

We next discuss the detail of the Power-EP procedure, that is how to iteratively update  $\lambda_t, \beta_t$  and  $\gamma_t$ ,

1. Deletion step: compute the cavity distribution

$$q^{\setminus i}(\mathbf{x}, f) \propto q(\mathbf{x}, f) / (\lambda_i(\mathbf{x}_{i-1})\beta_i(\mathbf{x}_i)\gamma_i(\mathbf{u}))^\alpha$$

2. Projection step:

$$q(\mathbf{x}, f) \leftarrow \operatorname{argmin}_{q(\mathbf{x}, f)} \text{KL}(\tilde{p}(\mathbf{x}, f) || q(\mathbf{x}, f))$$

$$\text{where } \tilde{p}(\mathbf{x}, f) = q^{\setminus i}(\mathbf{x}, f)p^\alpha(\mathbf{x}_i|f, \mathbf{x}_{i-1})$$

3. Update step:

$$\lambda_{i,\text{new}}(\mathbf{x}_{i-1})\beta_{i,\text{new}}(\mathbf{x}_i)\gamma_{i,\text{new}}(\mathbf{u}) \propto q(\mathbf{x}, f) / q^{\setminus i}(\mathbf{x}, f),$$

$$\lambda_i(\mathbf{x}_{i-1}) = \lambda_{i,\text{old}}^{1-\alpha}(\mathbf{x}_{i-1})\lambda_{i,\text{new}}(\mathbf{x}_{i-1}),$$

similarly for  $\beta_i(\mathbf{x}_i)$  and  $\gamma_i(\mathbf{u})$ .

When the factors are in the exponential family, the deletion and update steps above are simple as they only involve adding or subtracting (fractions of) corresponding natural parameters. The projection step is made tractable by a Gaussian projection which is nested in the computation of the log-normaliser of the tilted distribution,  $\log \mathcal{Z}_{\text{tilted},x,i}$ , where,

$$\log \mathcal{Z}_{\text{tilted}} = \int p^\alpha(\mathbf{x}_i|f, \mathbf{x}_{i-1}) p(f_{\neq \mathbf{u}}|\mathbf{u}) q^{\setminus i}(\mathbf{u}) q^{\setminus i}(\mathbf{x}_{i-1}) q^{\setminus i}(\mathbf{x}_i) df d\mathbf{x}_{i-1} d\mathbf{x}_i$$

The computation of  $\log \mathcal{Z}_{\text{tilted},x,i}$ , is crucial for the moment matching step of Power-EP, since the first and second moments can be computed from  $\log \mathcal{Z}_{\text{tilted},x,i}$  as discussed in appendix C. The first integral over  $f$  in the equation above is tractable since it only involves a marginal conditional distribution,  $p(f_i|\mathbf{u})$ , and the cavity  $q^{\setminus i}(\mathbf{u})$ . Following (Girard et al., 2003; Deisenroth and Mohamed, 2012; Bui et al., 2016), we approximate the second integral w.r.t.  $\mathbf{x}_{i-1}$  by a Gaussian distribution over  $\mathbf{x}_i$  with the following mean and variance,

$$\tilde{m}_i = \mathbf{E}_1 \mathbf{A} \quad (138)$$

$$\tilde{v}_i = \sigma_x^2/\alpha + w_i - \tilde{m}_i^2 \quad (139)$$

where  $w_i = \mathbf{E}_0 + \text{tr}(\mathbf{B}\mathbf{E}_2)$ ,  $\mathbf{E}_0 = \mathbf{E}_{q^{\setminus i}(\mathbf{x}_{i-1})}[K_{f_i,f_i}]$ ,  $\mathbf{E}_1 = \mathbf{E}_{q^{\setminus i}(\mathbf{x}_{i-1})}[\mathbf{K}_{f_i,\mathbf{u}}]$ ,  $\mathbf{E}_2 = \mathbf{E}_{q^{\setminus i}(\mathbf{x}_{i-1})}[\mathbf{K}_{\mathbf{u},f_i}\mathbf{K}_{f_i,\mathbf{u}}]$ ,  $\mathbf{A} = \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}}^{\setminus i}$  and  $\mathbf{B} = \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}(\mathbf{V}_{\mathbf{u}}^{\setminus i} + \mathbf{m}_{\mathbf{u}}^{\setminus i}\mathbf{m}_{\mathbf{u}}^{\setminus i,T})\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} - \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}$ . The equations above require the expectations of the kernel matrix under a Gaussian distribution over the inputs, which are analytically tractable for widely used kernels such as exponentiated quadratic, linear or a more general class of spectral mixture kernels (Titsias and Lawrence, 2010). The third integral w.r.t  $\mathbf{x}_i$  is now straightforward as it is the normaliser of a convolution of two Gaussian distributions. In spite of this Gaussian projection, as  $\alpha \rightarrow 0$ , we recover McHutchon's variational treatment with an additional factorised assumption over  $x_{1:T}$ ,  $q(\mathbf{x}_{1:T}) = \prod_t q(\mathbf{x}_t)$  (McHutchon, 2014). We will show this equivalence in the next section.

## F.2 The approximate marginal likelihood

Similar to the regression and classification case, we can obtain an approximate marginal likelihood as follows,

$$\begin{aligned} \mathcal{F} = & \phi_{\text{post},f} + \phi_{\text{post},\mathbf{x}_0} - \phi_{\text{prior},f} - \phi_{\text{prior},\mathbf{x}_0} + \left[ \frac{1}{\alpha} \sum_t \log \mathcal{Z}_{\text{tilted},x,t} + \phi_{\text{cav},x,t} - \phi_{\text{post}} \right] \\ & + \left[ \frac{1}{\alpha} \sum_t \log \mathcal{Z}_{\text{tilted},y,t} + \phi_{\text{cav},y,t} - \phi_{\text{post}} \right]. \end{aligned} \quad (140)$$

The above expression is not analytically tractable due to the difficult term  $\log \mathcal{Z}_{\text{tilted},x,t}$ ; however, this can be approximated using the Gaussian projection discussed in the previous section. For simplicity, we will consider an one dimensional hidden variable case. Letting  $q^{\setminus t}(x_t) = \mathcal{N}(x_t; m_t, v_t)$  and using eqns. eq. (138) and 139, we have,

$$2 \log \mathcal{Z}_{\text{tilted},x,t} \approx \log \frac{2\pi\sigma_x^2/\alpha}{(2\pi\sigma_x^2)^\alpha} - \log(2\pi) - \log\left(\frac{\sigma_x^2}{\alpha} + w_t - \tilde{m}_t^2 + v_t\right) - \frac{(m_t - \tilde{m}_t)^2}{\frac{\sigma_x^2}{\alpha} + w_t - \tilde{m}_t^2 + v_t} \quad (141)$$

$$= -\log\left(1 + \frac{\alpha(w_t - \tilde{m}_t^2 + v_t)}{\sigma_x^2}\right) - \alpha \log(2\pi\sigma_x^2) - \frac{\alpha(m_t - \tilde{m}_t)^2}{\sigma_x^2 + \alpha(w_t - \tilde{m}_t^2 + v_t)} \quad (142)$$

We divide the above by  $\alpha$  and obtain its limits as  $\alpha \rightarrow 0$ ,

$$\lim_{\alpha \rightarrow 0} \frac{1}{\alpha} \log \mathcal{Z}_{\text{approx,tilted},x,t} = -\frac{1}{2} \log(2\pi\sigma_x^2) - \frac{1}{2} \frac{(w_t - \tilde{m}_t^2 + v_t)}{\sigma_x^2} - \frac{1}{2} \frac{(m_t - \tilde{m}_t)^2}{\sigma_x^2} \quad (143)$$

$$= -\frac{1}{2} \log(2\pi\sigma_x^2) - \frac{1}{2} \frac{m_t^2 - 2m_t\tilde{m}_t + w_t + v_t}{\sigma_x^2} \quad (144)$$

Despite that we have used the Gaussian approximation, the limit above is exactly the expected log likelihood,  $\mathcal{F}_t = \langle \log p(x_t | f, x_{t-1}) \rangle_{q(f)q(x_t)q(x_{t-1})}$ , as appeared in the variational lower bound,

$$\mathcal{F}_t = \left\langle -\frac{1}{2} \log(2\pi\sigma_x^2) - \frac{1}{2} \frac{(x_t - f(x_{t-1}))^2}{\sigma_x^2} \right\rangle_{q(f)q(x_t)q(x_{t-1})} \quad (145)$$

$$= -\frac{1}{2} \log(2\pi\sigma_x^2) - \frac{1}{2} \frac{\langle x_t^2 \rangle_{q(x_t)} - 2\langle x_t \rangle_{q(x_t)} \langle f(x_{t-1}) \rangle_{q(f)q(x_{t-1})} + \langle f(x_{t-1})^2 \rangle_{q(f)q(x_{t-1})}}{\sigma_x^2} \quad (146)$$

$$= -\frac{1}{2} \log(2\pi\sigma_x^2) - \frac{1}{2} \frac{m_t^2 - 2m_t\tilde{m}_t + w_t + v_t}{\sigma_x^2}. \quad (147)$$

## Appendix G. Extra experimental results

### G.1 Comparison between various $\alpha$ values on a toy regression problem

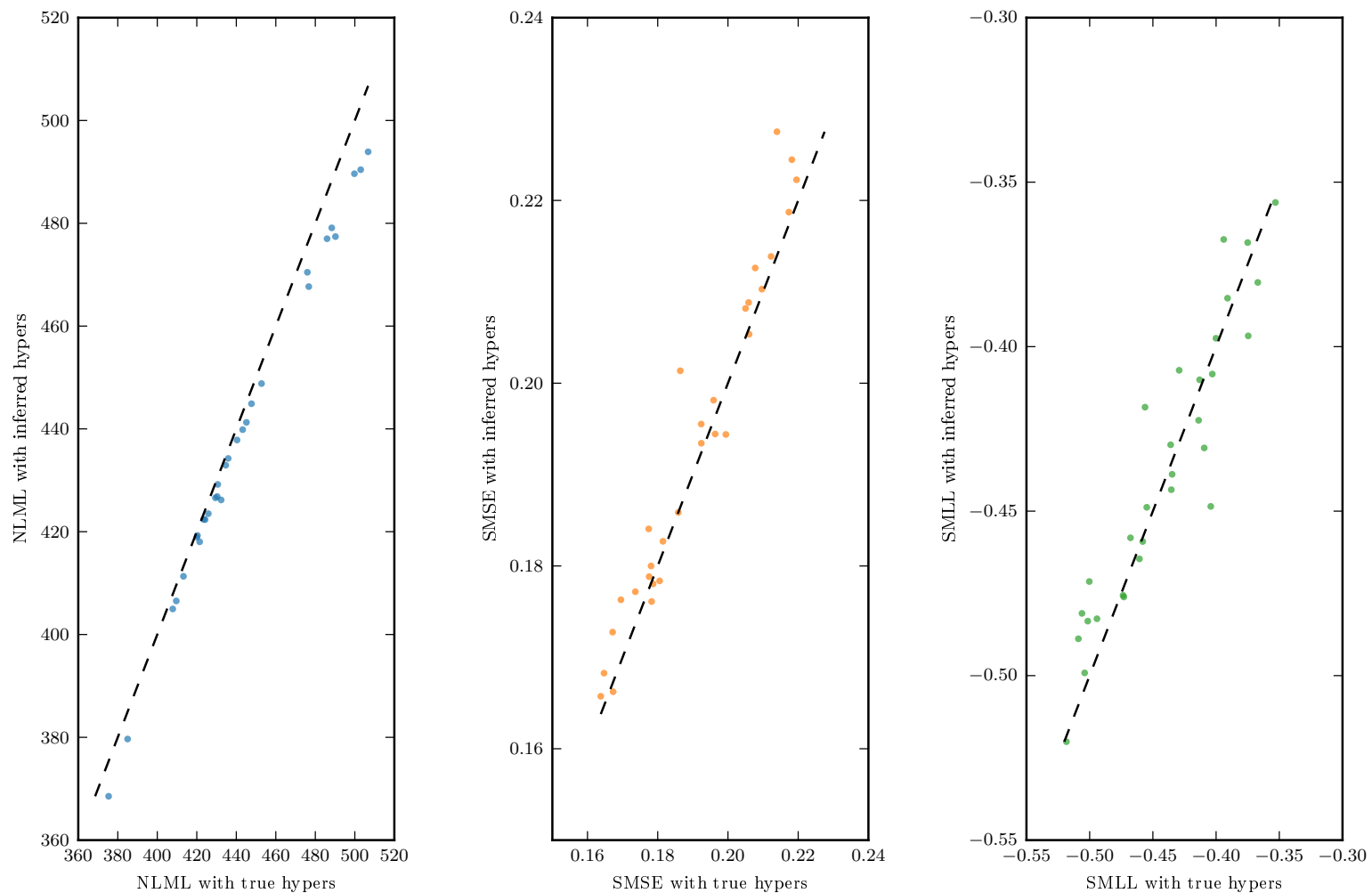


Figure 9: Results on a toy regression problem: Negative log-marginal likelihood, mean squared error and mean log-loss on the test set for full Gaussian process regression on synthetic datasets with *true* hyper-parameters and hyper-parameters obtained by type-2 ML. Each dot is one trial, i.e. one synthetic dataset. The results demonstrate that type-2 maximum likelihood on hyper-parameters works well, despite being a little confident on the log-marginal likelihood on the train set.



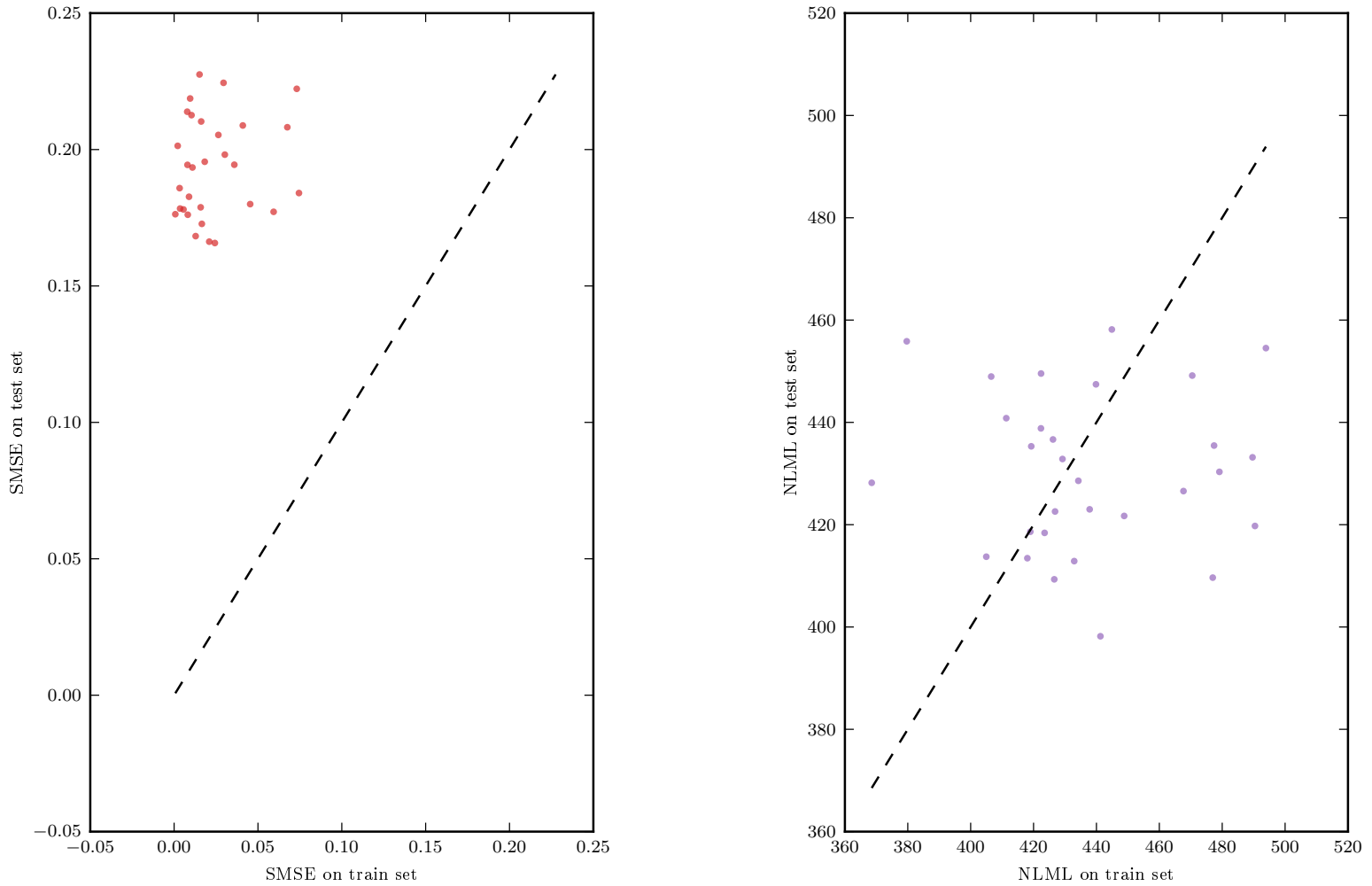


Figure 10: Results on a toy regression problem with 500 training points: Mean squared error and log-likelihood on train and test sets on synthetic datasets with hyper-parameters obtained by type-2 ML. In this example, the test error is higher than the training error, as measured by the mean squared error, because the test points and training points are relatively far apart, making the prediction task on the training set easier (interpolation) than on the test set (extrapolation). This is consistent with the results with more training points, shown in fig. 11.

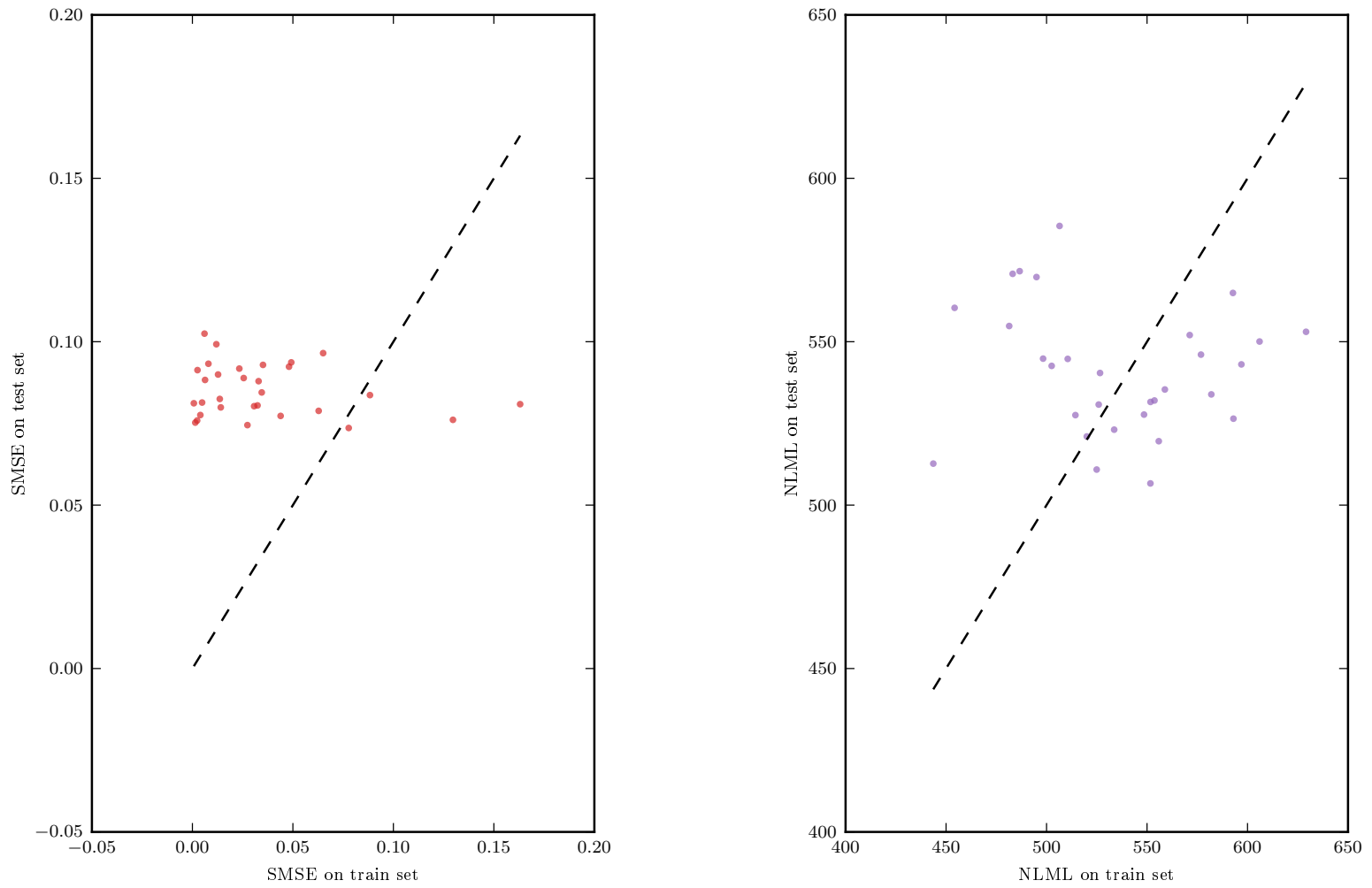


Figure 11: Results on a toy regression problem with 1000 training points: Mean squared error and log-likelihood on train and test sets on synthetic datasets with hyper-parameters obtained by type-2 ML. See fig. 10 for a discussion.

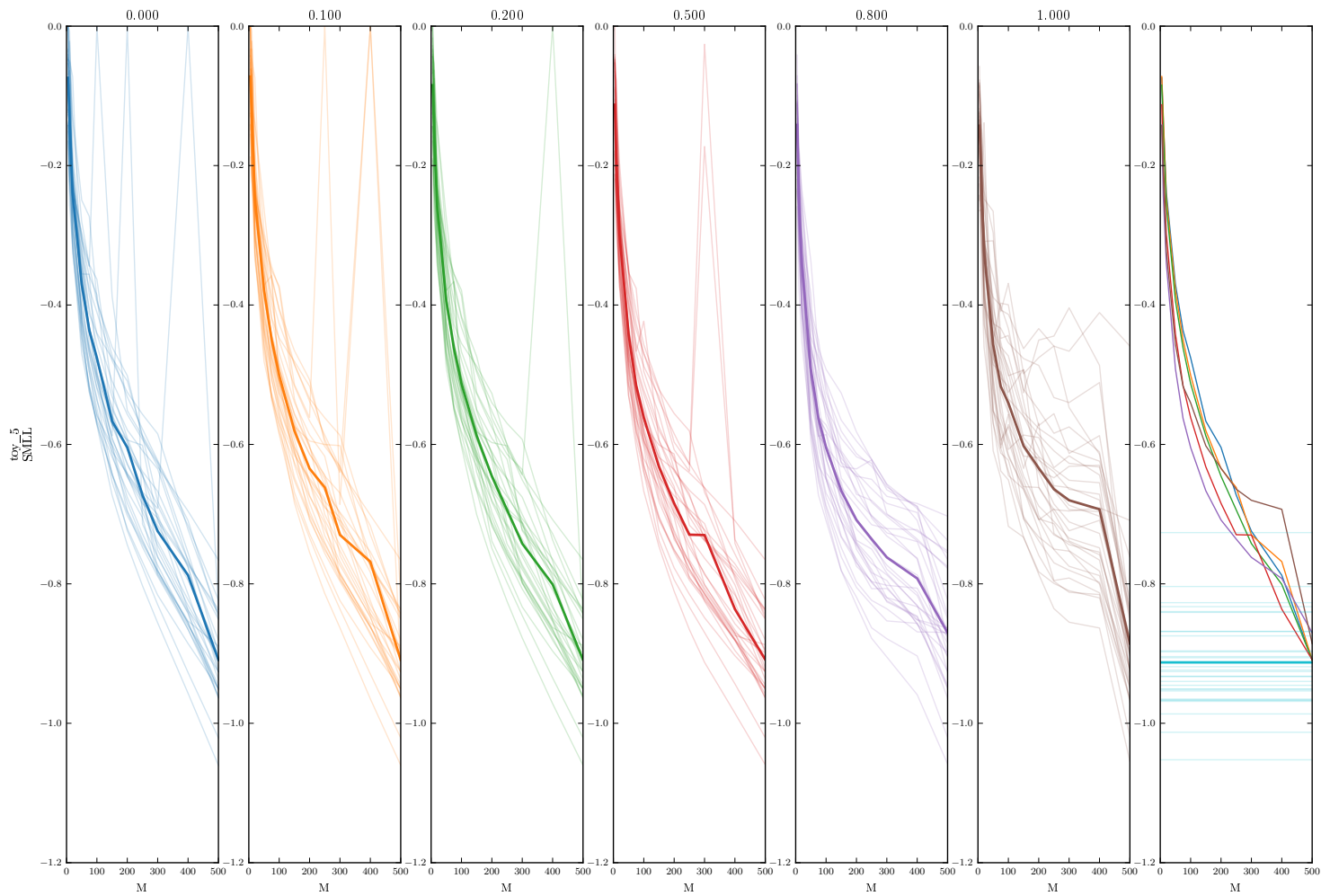


Figure 12: Results on a toy regression problem: Standardised mean log-loss on the test set for various values of  $\alpha$  and various number of pseudo-points  $M$ . Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various  $\alpha$ , and the results using GP regression.

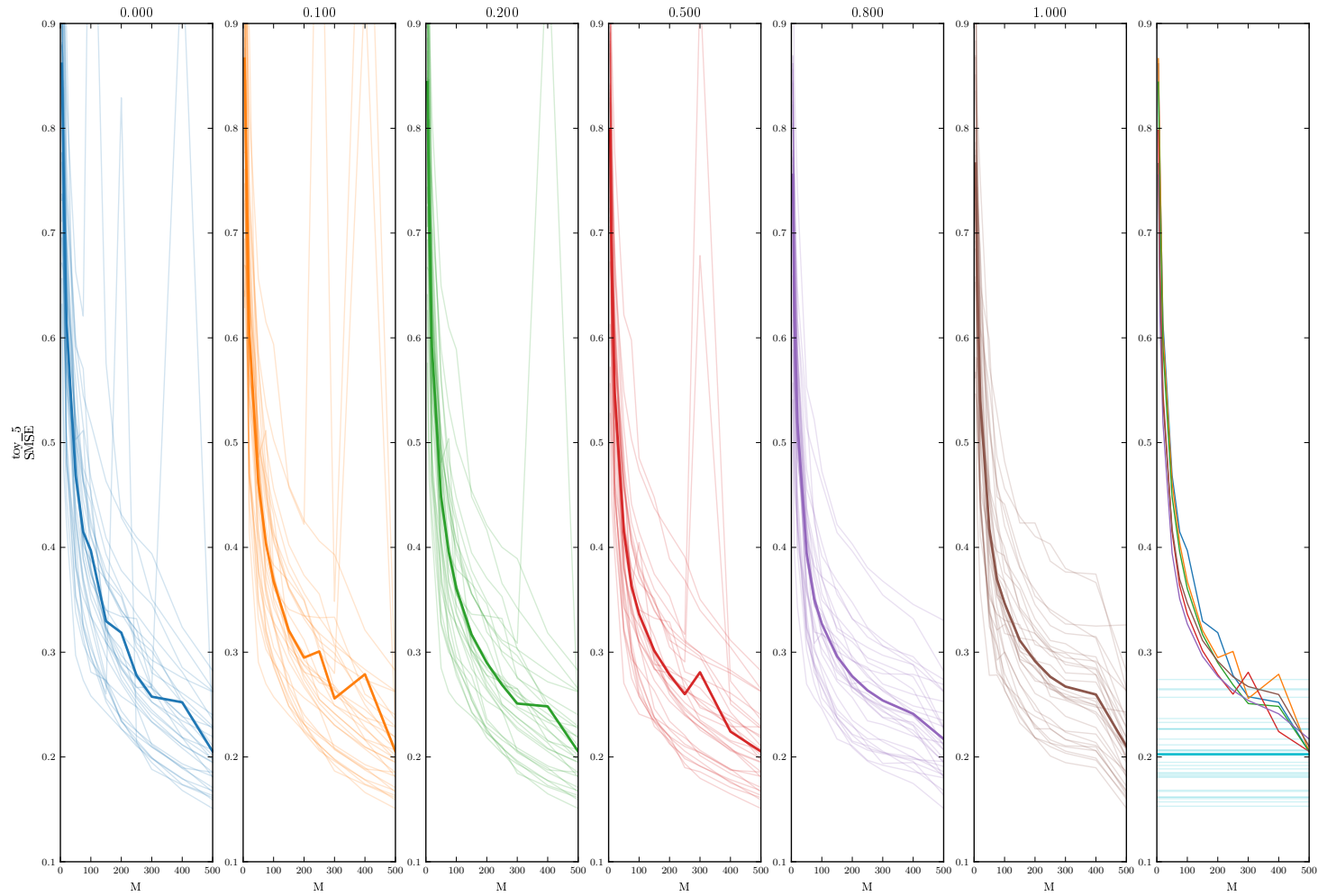


Figure 13: Results on a toy regression problem: Standardised mean squared error on the test set for various values of  $\alpha$  and various number of pseudo-points  $M$ . Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various  $\alpha$ , and the results using GP regression.

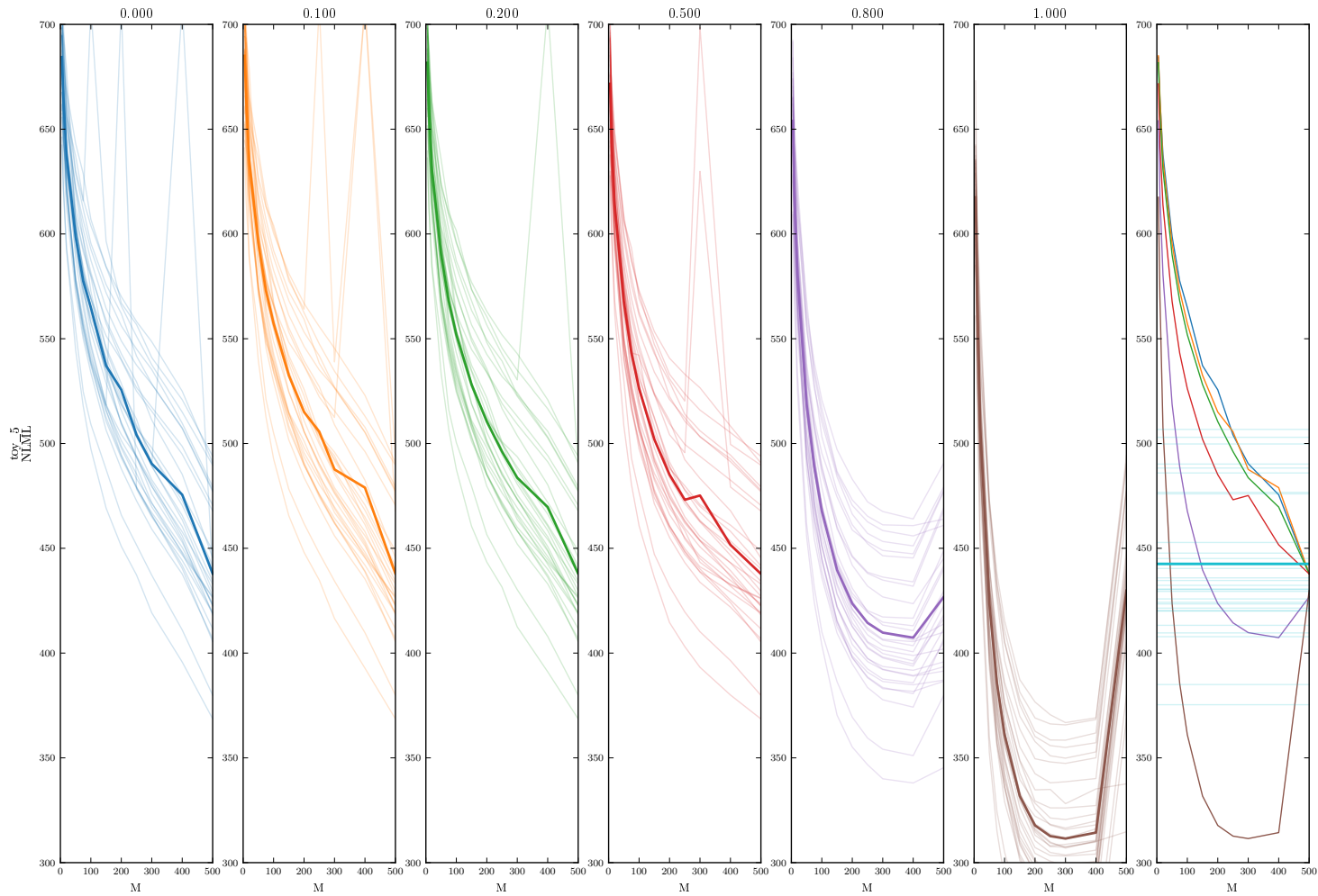


Figure 14: Results on a toy regression problem: The negative log marginal likelihood of the training set after training for various values of  $\alpha$  and various number of pseudo-points  $M$ . Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various  $\alpha$ , and the results using GP regression. Power EP with  $\alpha$  close to 1 over-estimates the marginal-likelihood.

## G.2 Real-world regression

We include the details of the regression datasets in table 1 and several comparisons of  $\alpha$  values in figs. 18 to 23.

Dataset	N train/test	D
boston	455/51	14
concrete	927/103	9
energy	691/77	9
kin8nm	7373/819	9
naval	10741/1193	18
yacht	277/31	7
power	8611/957	5
red wine	1439/160	12

Table 1: Regression datasets

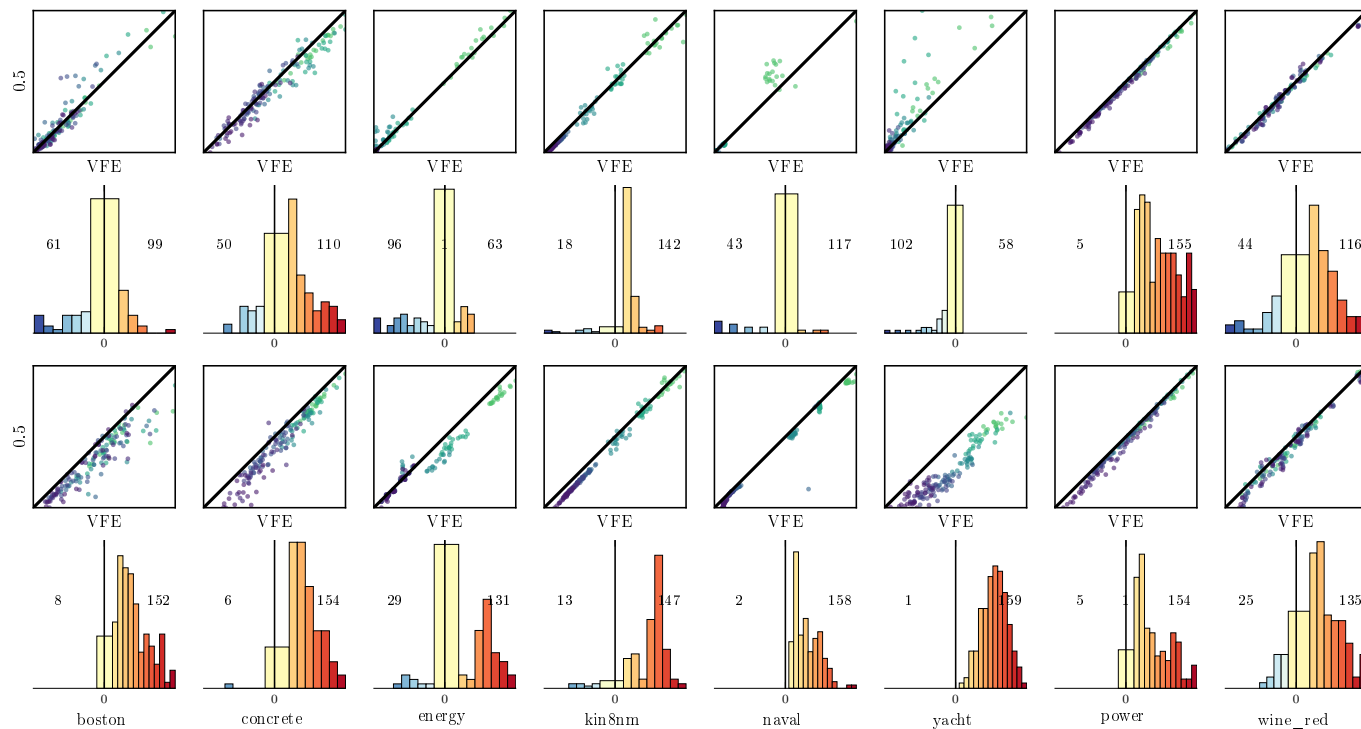


Figure 15: A comparison between Power-EP with  $\alpha = 0.5$  and VFE on several regression datasets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). The scatter plots show the performance of Power-EP ( $\alpha = 0.5$ ) vs VFE. Each point is one split and points with lighter colours are runs with big  $M$ . Points that stay below the diagonal line show  $\alpha = 0.5$  is better than VFE. The plots right underneath the scatter plots show the histogram of the difference between methods. Red means  $\alpha = 0.5$  is better than VFE.

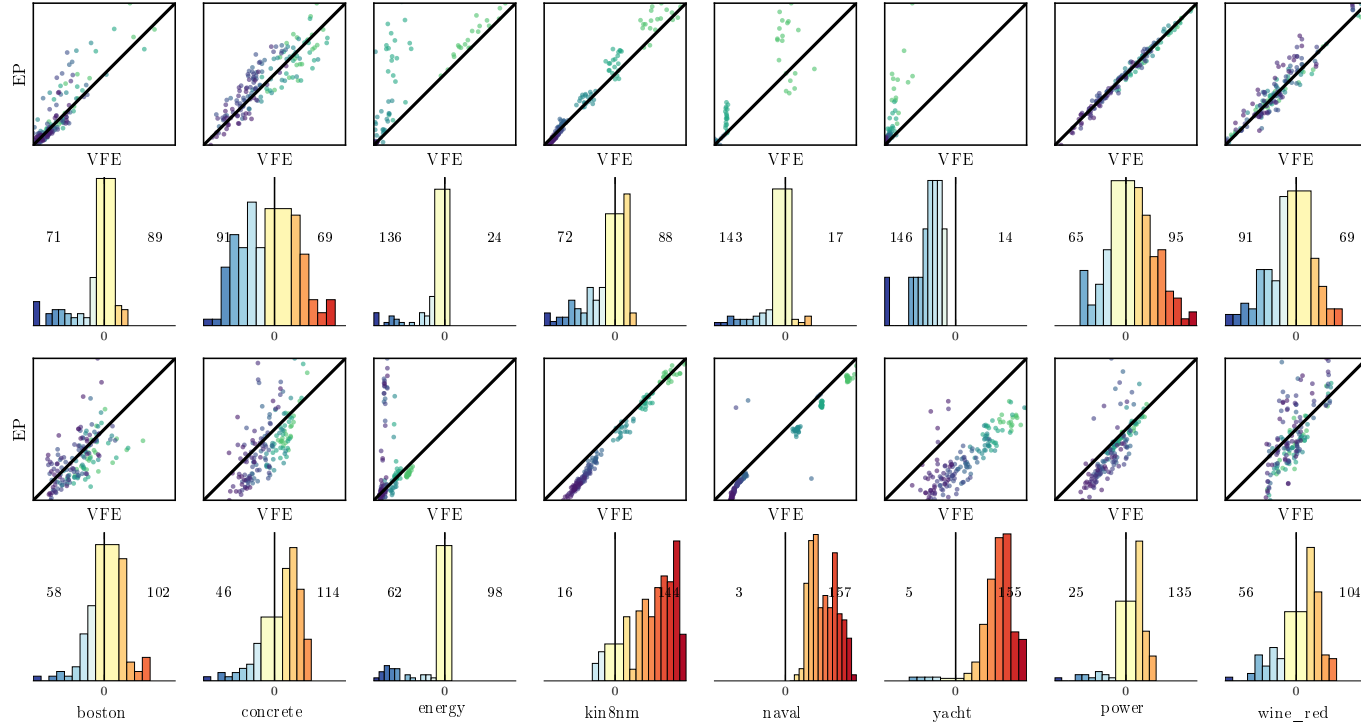


Figure 16: A comparison between EP and VFE on several regression datasets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). See fig. 15 for more details about the plots.



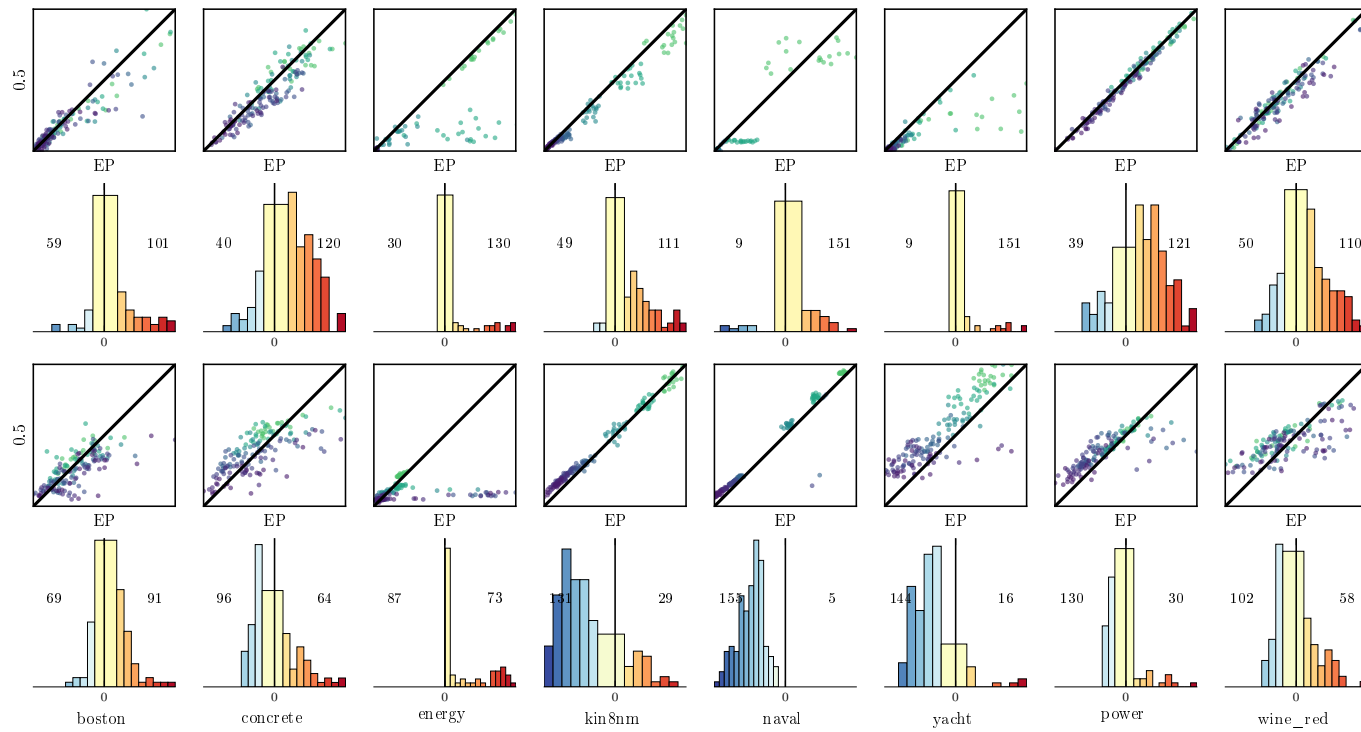


Figure 17: A comparison between Power-EP with  $\alpha = 0.5$  and EP on several regression datasets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). See fig. 15 for more details about the plots.

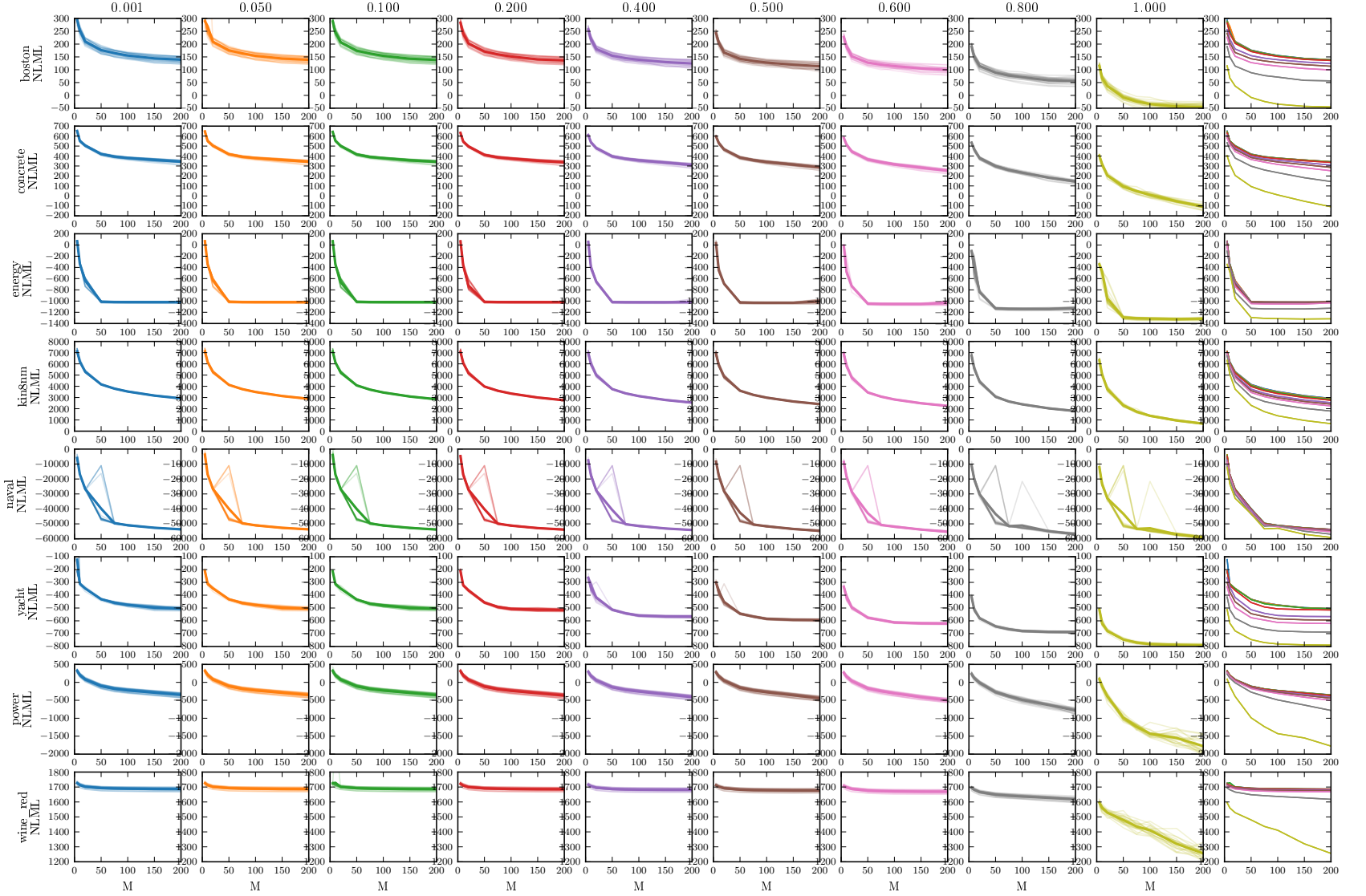


Figure 18: Results on real-world regression problems: Negative training log-marginal likelihood for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ . Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various  $\alpha$  for comparison. Lower is better [however, lower could mean overestimation].

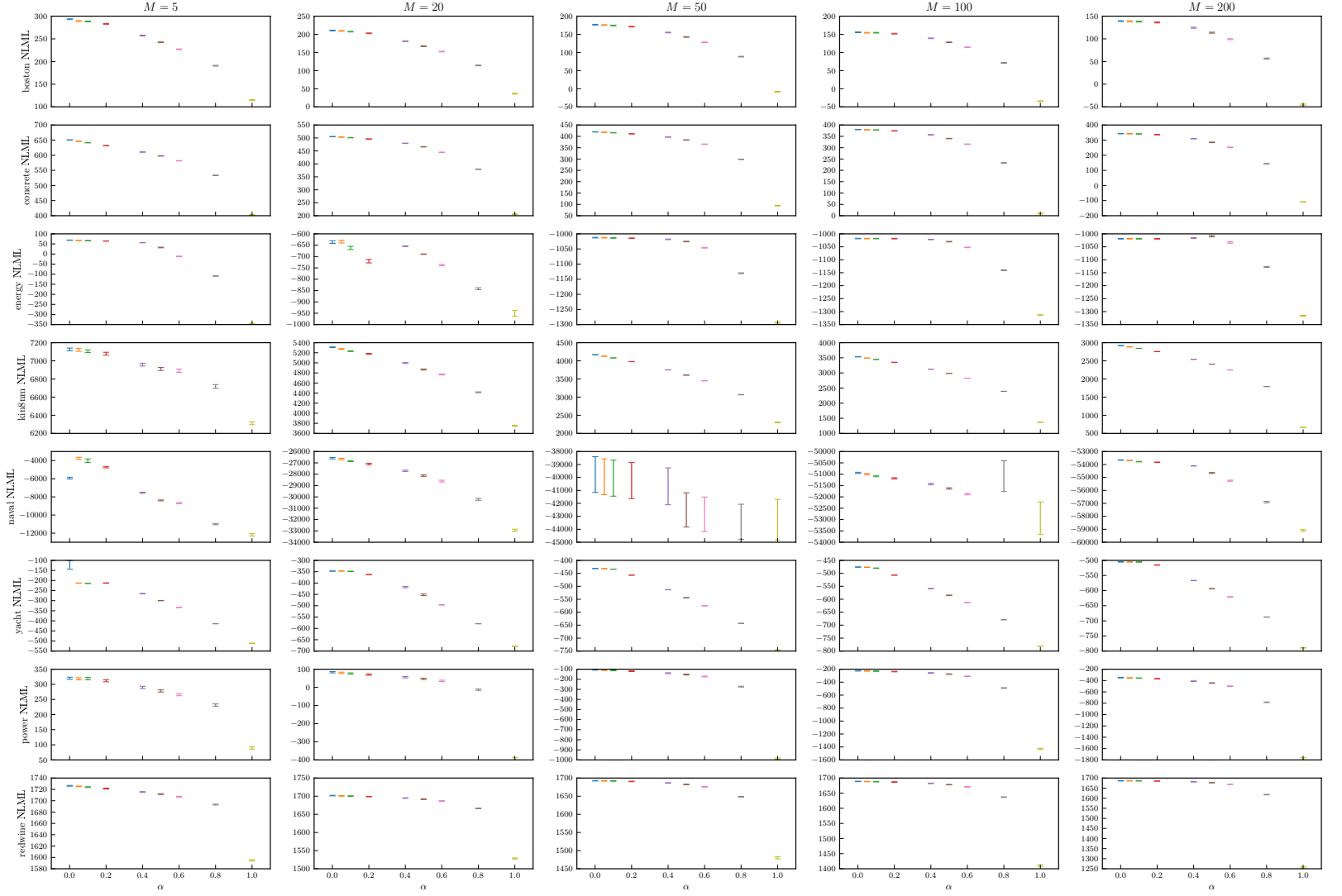


Figure 19: Results on real-world regression problems: Negative training log-marginal likelihood for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ , averaged over 20 splits. Lower is better [however, lower could mean overestimation].

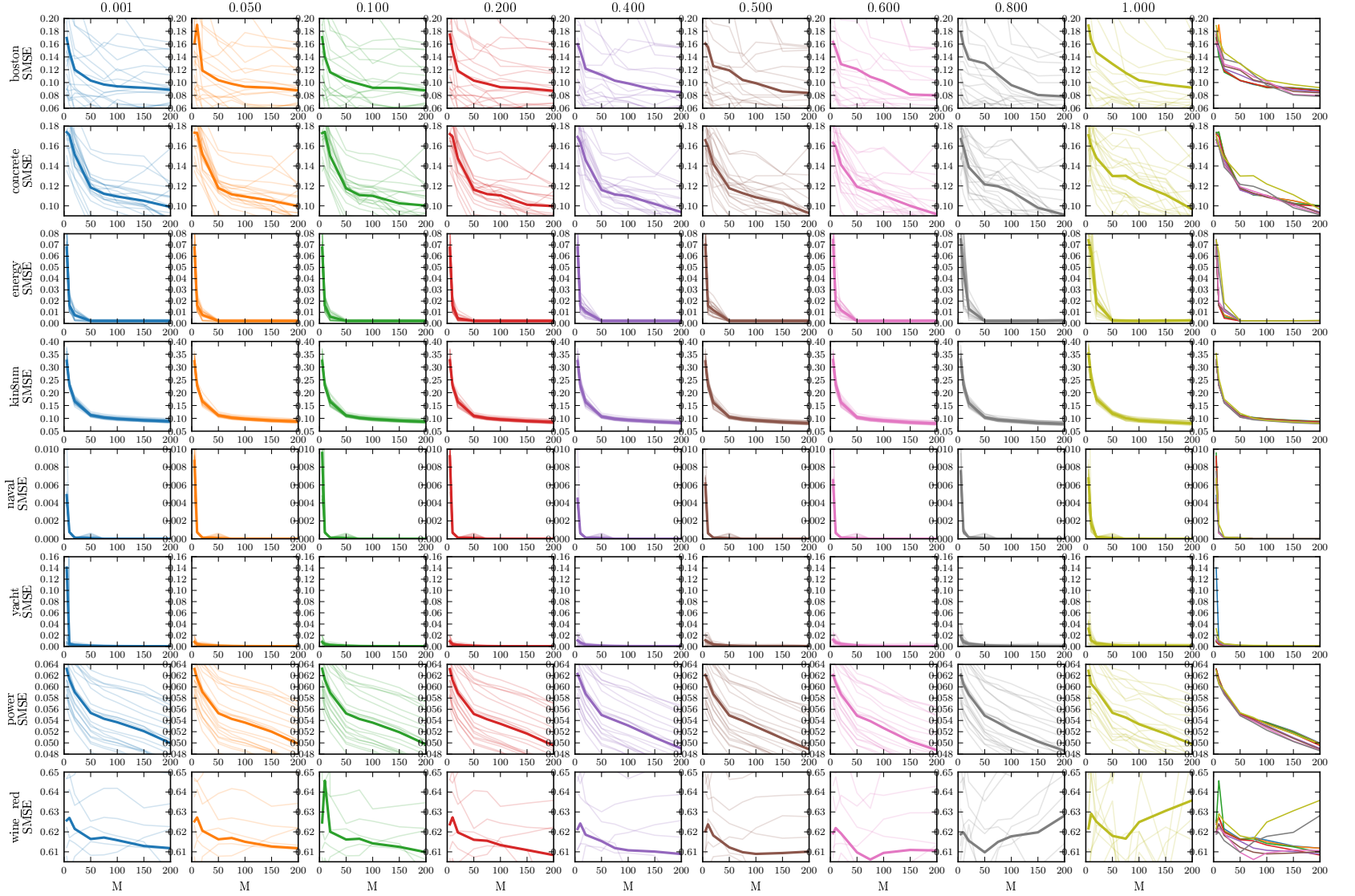


Figure 20: Results on real-world regression problems: Standardised mean squared error on the test set for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ . Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various  $\alpha$  for comparison. Lower is better.

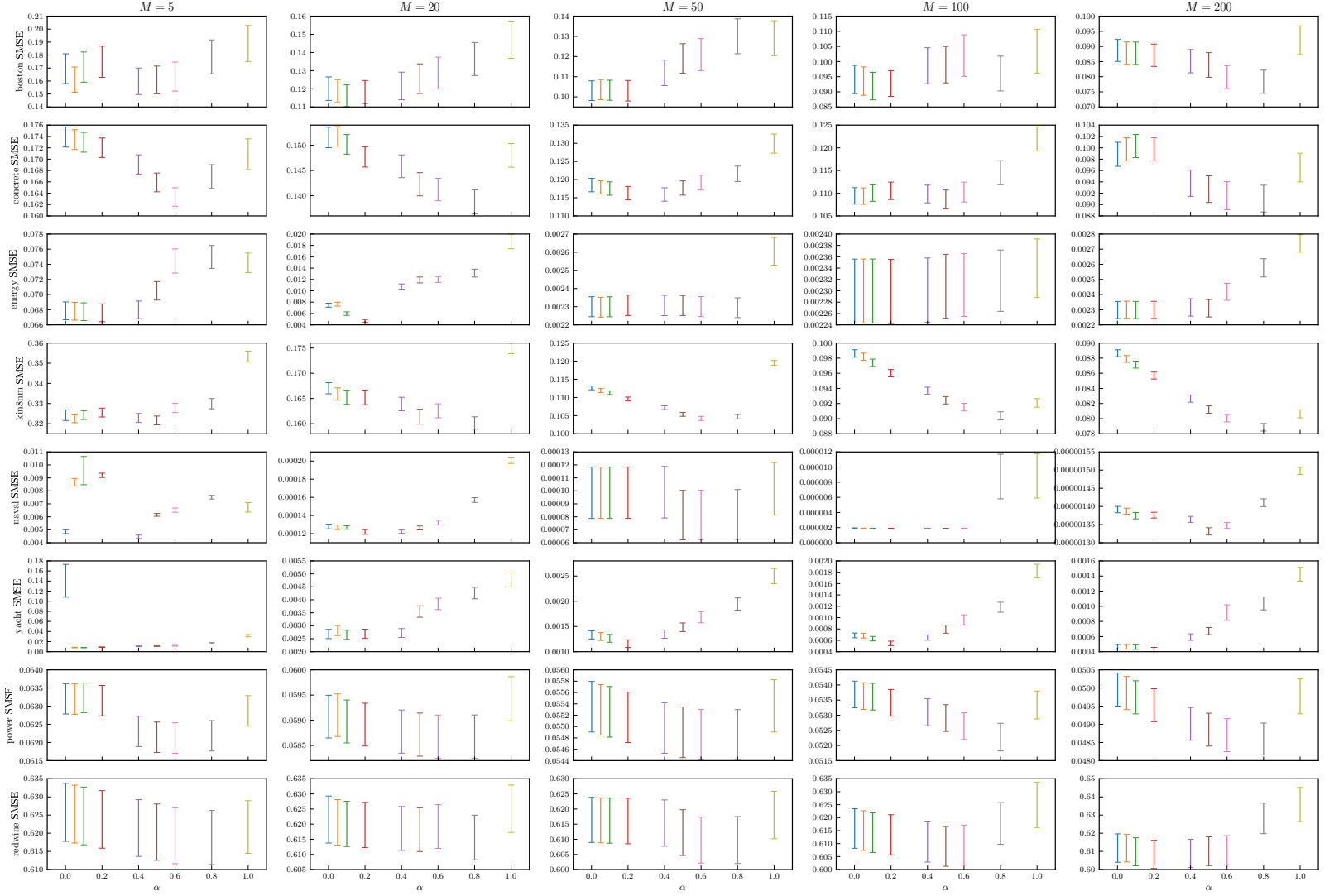


Figure 21: Results on real-world regression problems: Standardised mean squared error on the test set for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ , averaged over 20 splits. Lower is better.

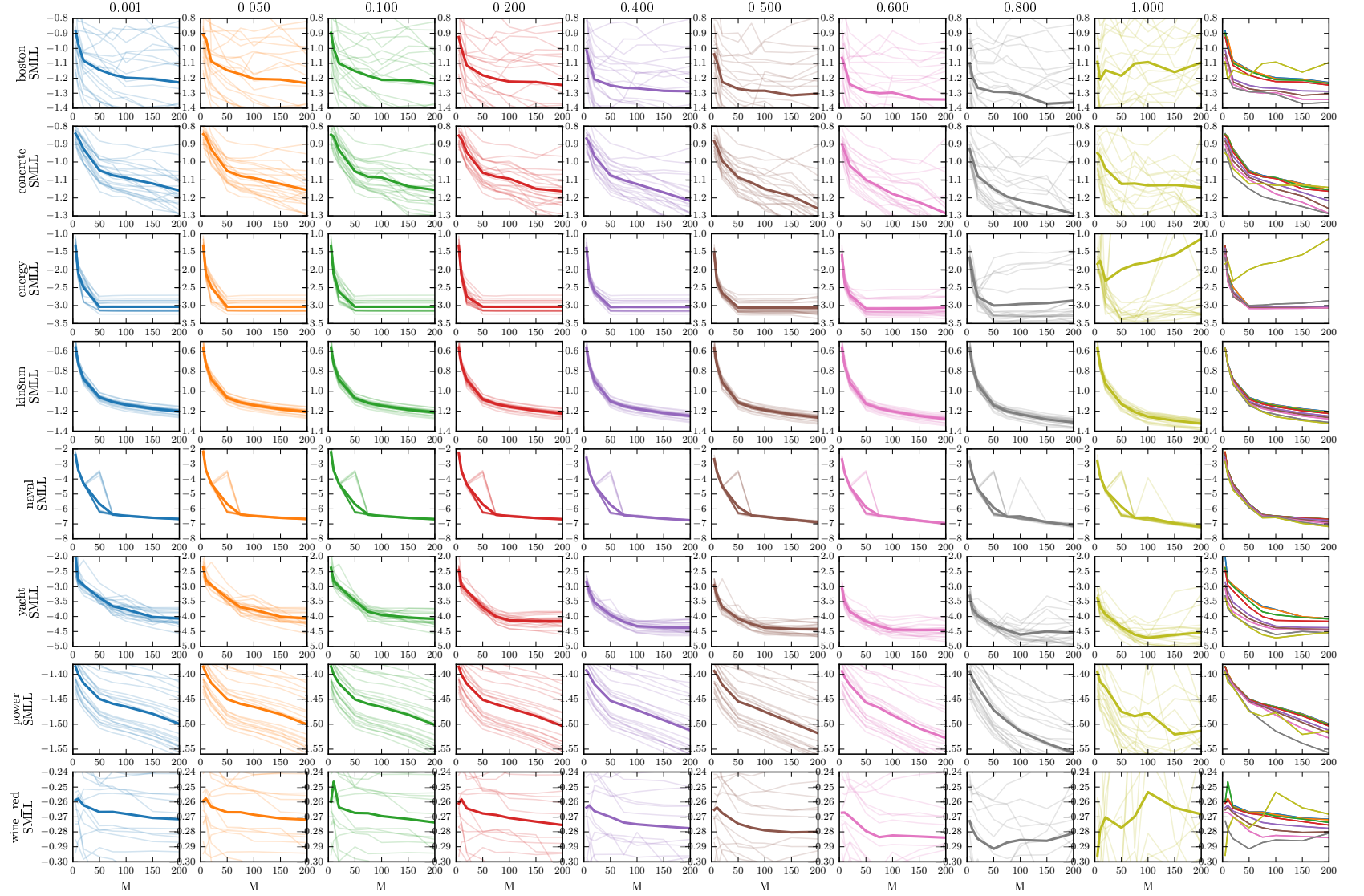


Figure 22: Results on real-world regression problems: Standardised mean log loss on the test set for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ . Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various  $\alpha$  for comparison. Lower is better.

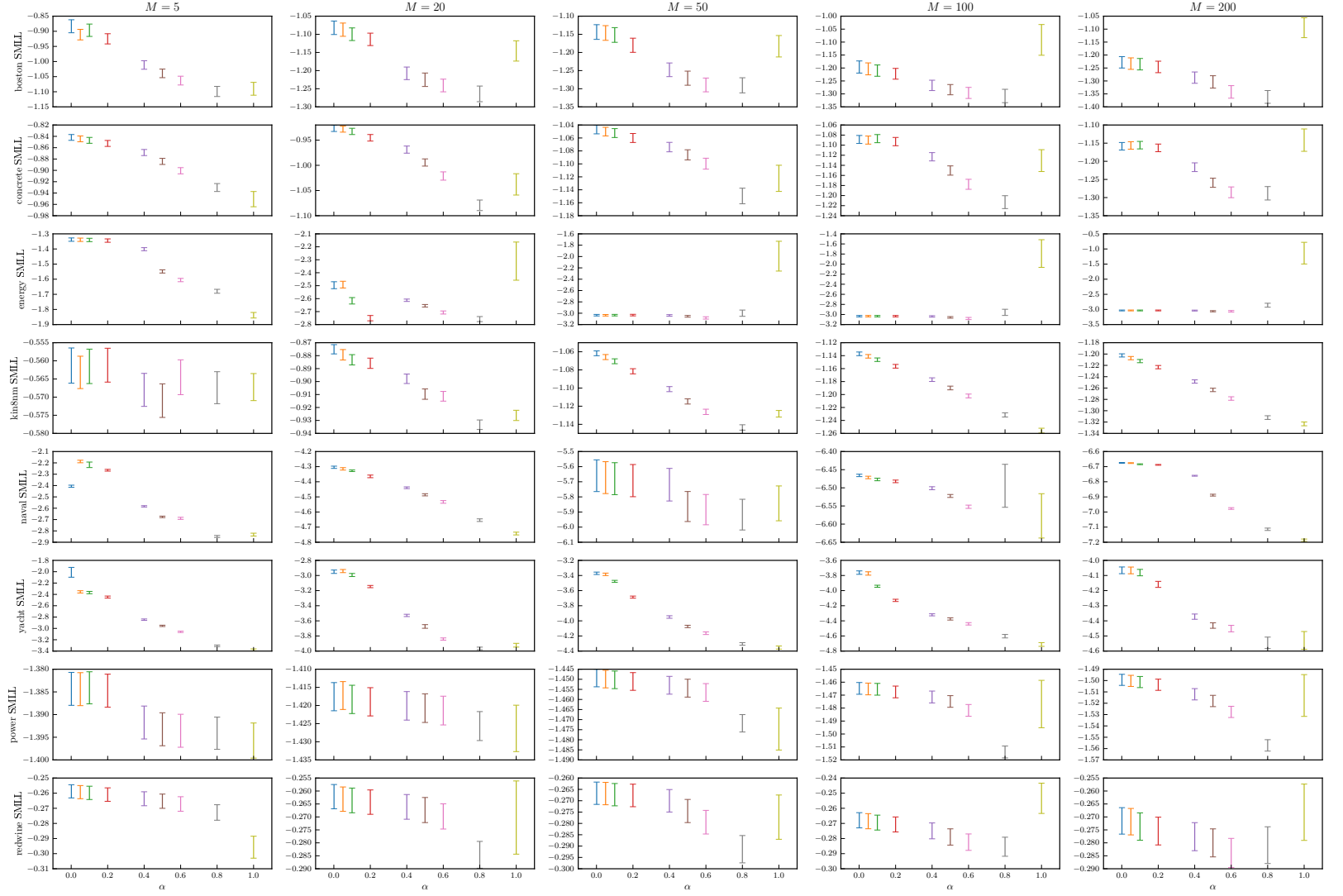


Figure 23: Results on real-world regression problems: Standardised mean log loss on the test set for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ , averaged over 20 splits. Lower is better.



### G.3 Real-world classification

It was demonstrated in (Hernández-Lobato and Hernández-Lobato, 2016; Hensman et al., 2015) that, once optimised, the pseudo points tend to concentrate around the decision boundary for VFE, and spread out to cover the data region in EP. Figure 24 illustrates the same effect as  $\alpha$  goes from close to 0 (VFE) to 1 (EP).

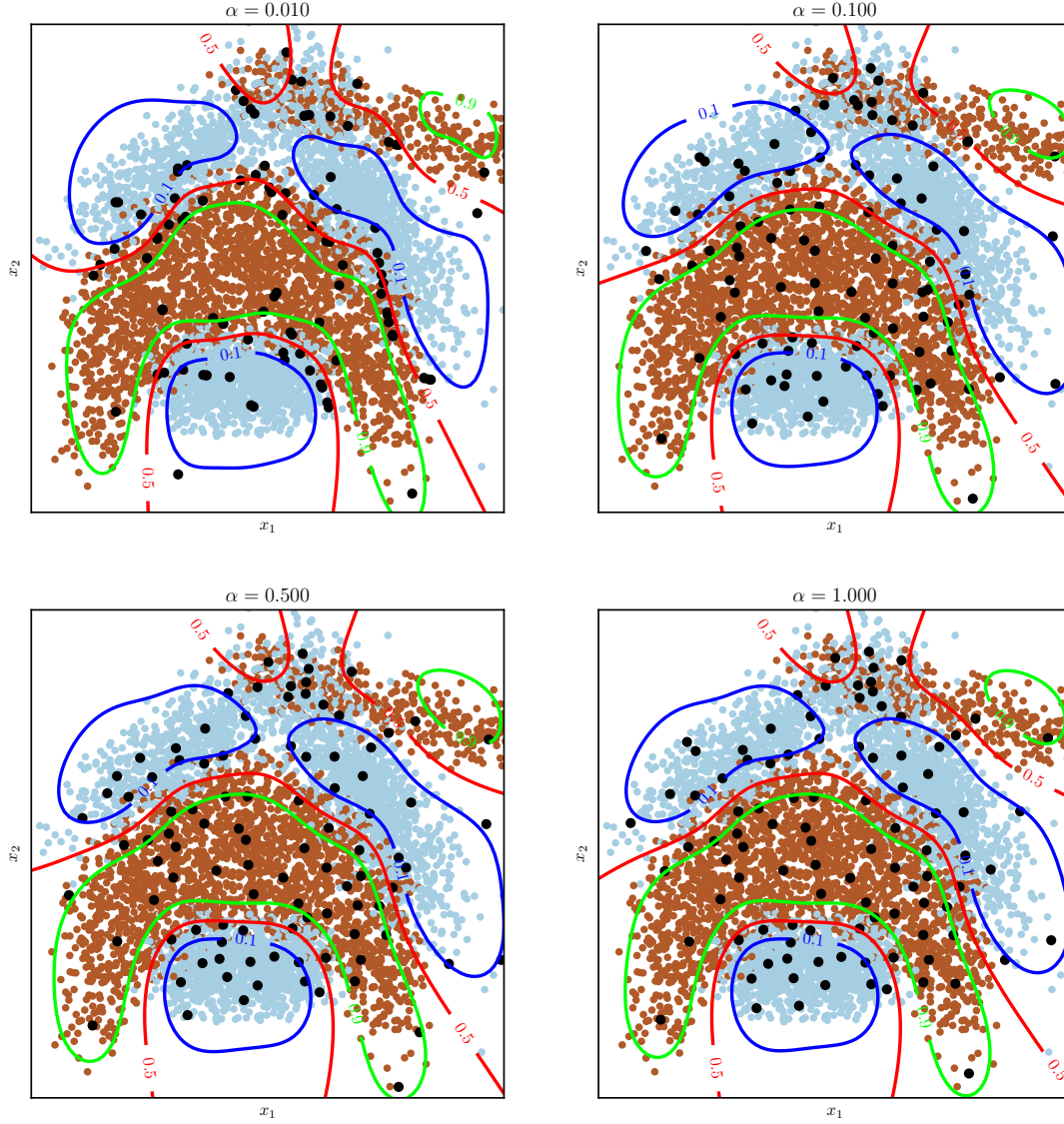


Figure 24: The locations of pseudo data points vary with  $\alpha$ . Best viewed in colour.

We include the details of the classification datasets in table 2 and several comparisons of  $\alpha$  values in figs. 28 to 31.



Dataset	N train/test	D
australian	621/69	15
breast	614/68	11
crabs	180/20	7
iono	315/35	35
pima	690/77	9
sonar	186/21	61

Table 2: Classification datasets

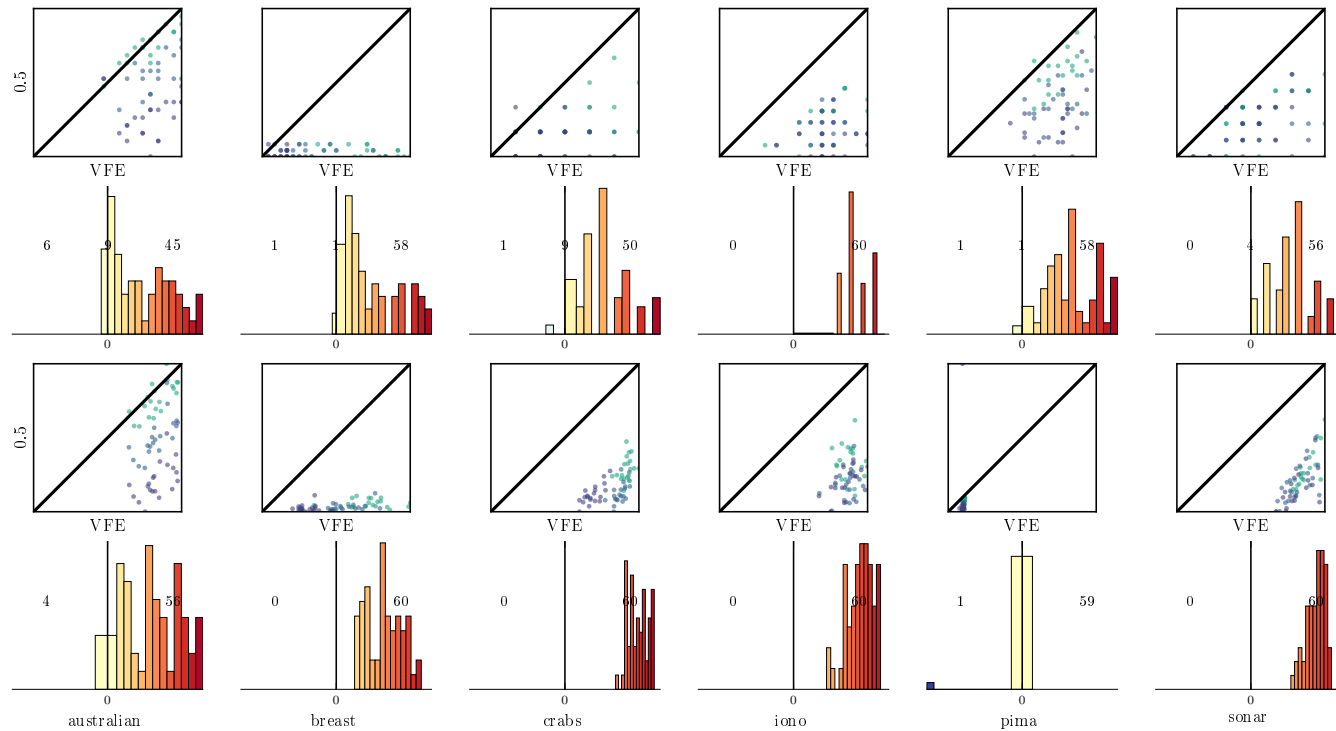


Figure 25: A comparison between Power-EP with  $\alpha = 0.5$  and VFE on several classification datasets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See fig. 15 for more details about the plots.

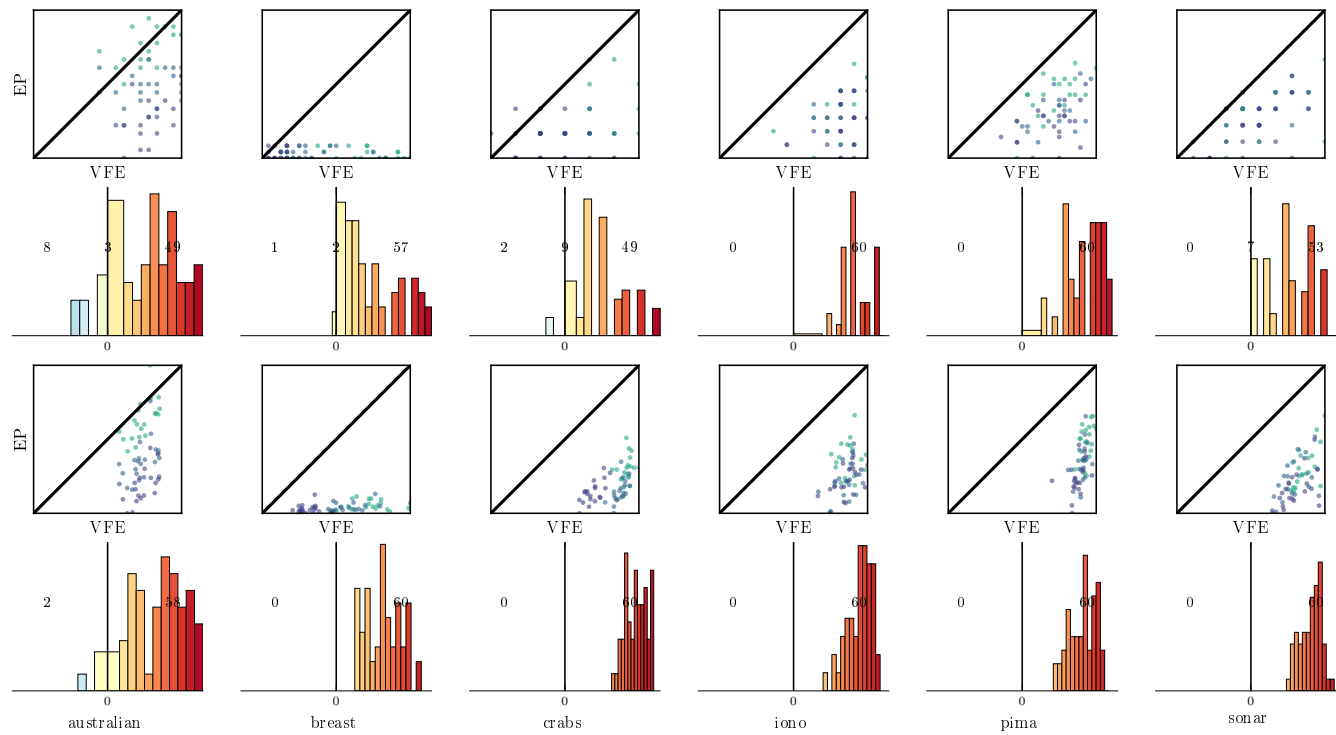


Figure 26: A comparison between EP and VFE on several classification datasets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See fig. 15 for more details about the plots.

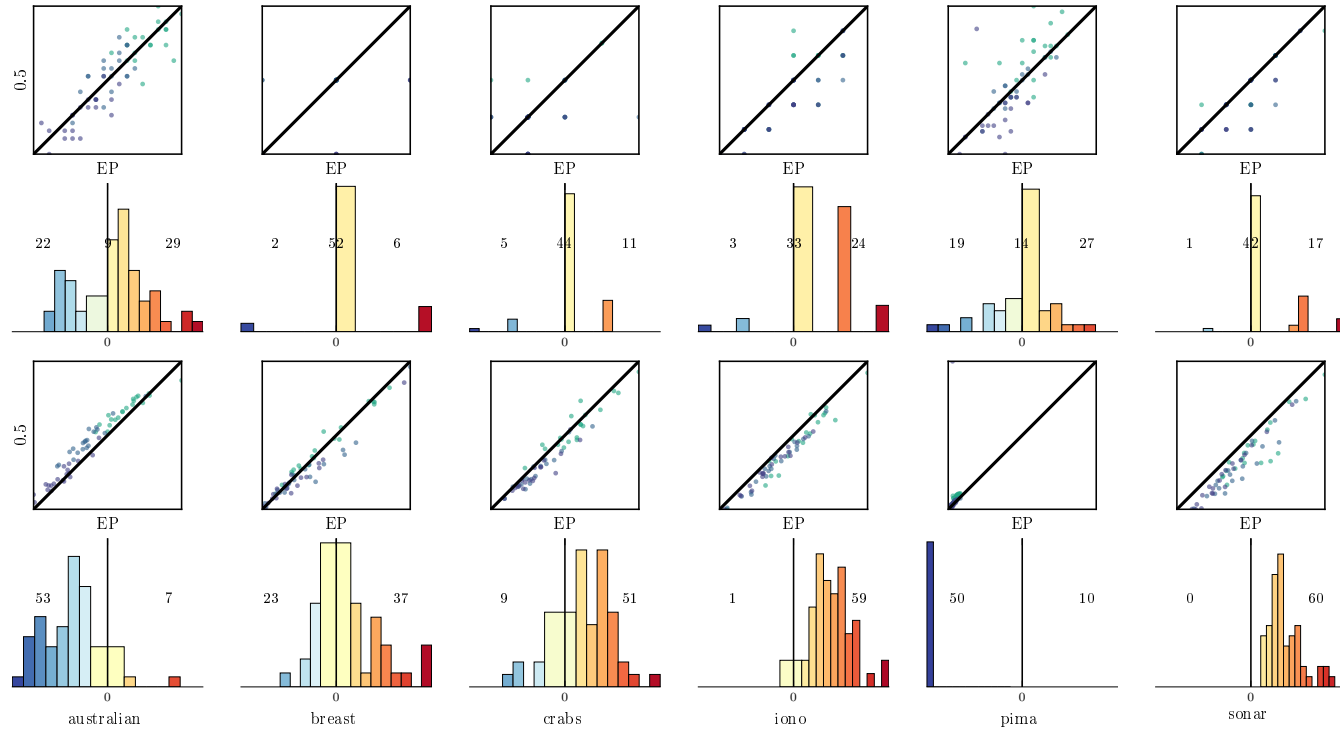


Figure 27: A comparison between Power-EP with  $\alpha = 0.5$  and EP on several classification datasets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See fig. 15 for more details about the plots.

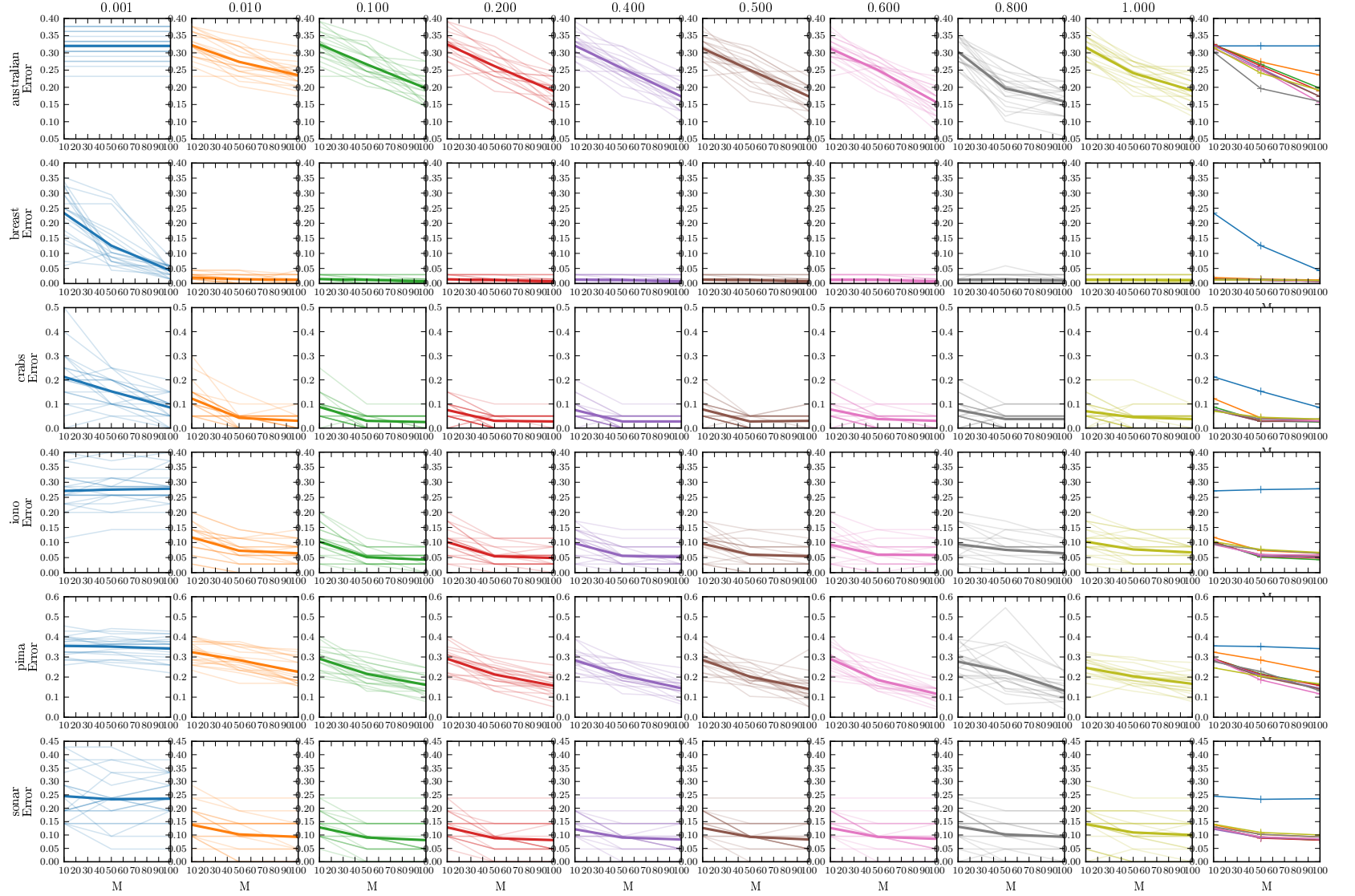


Figure 28: Results on real-world classification problems: Classification error rate on the test set for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ . Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various  $\alpha$  for comparison. Lower is better.

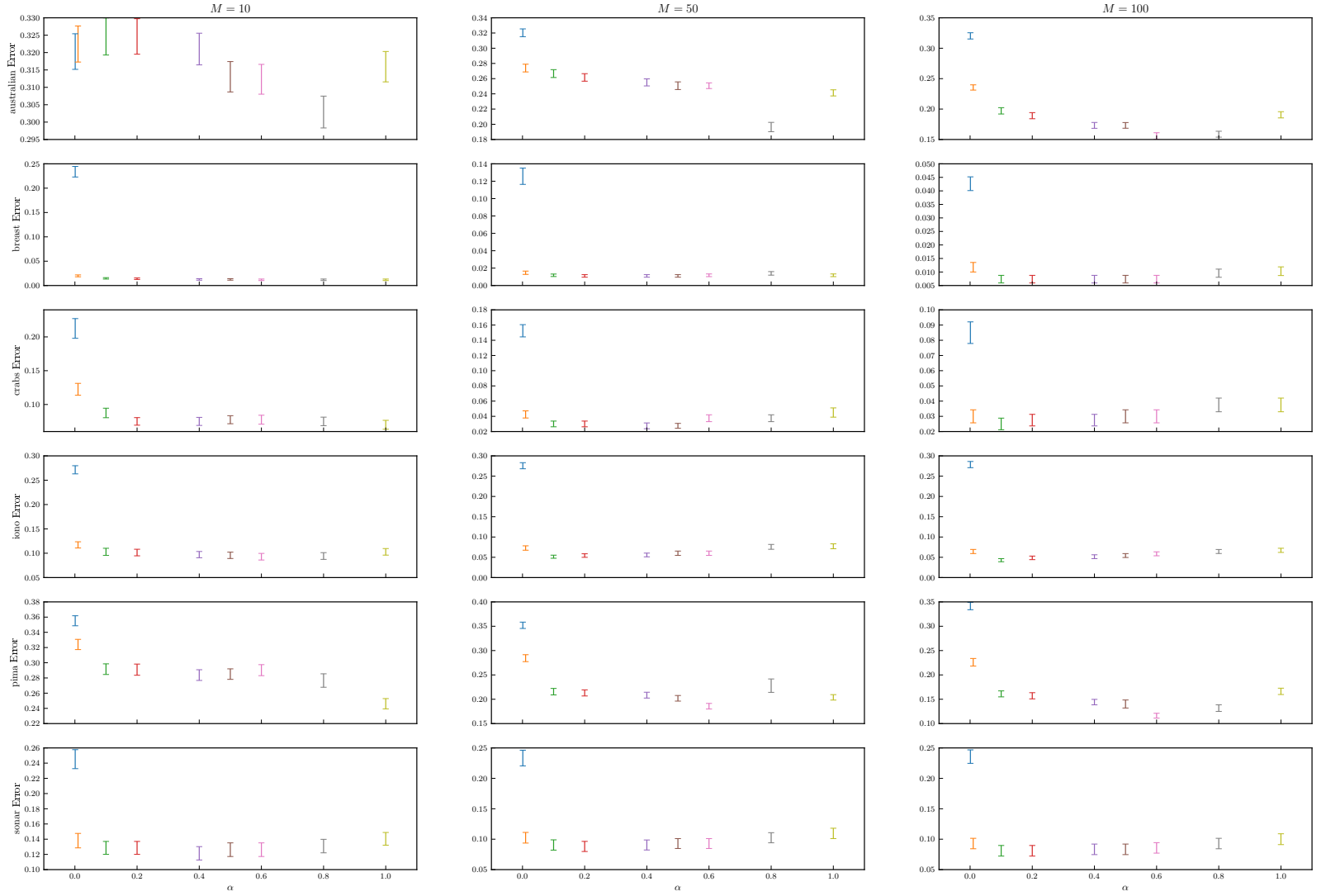


Figure 29: Results on real-world classification problems: Classification error rate on the test set for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ , averaged over 20 splits. Lower is better.

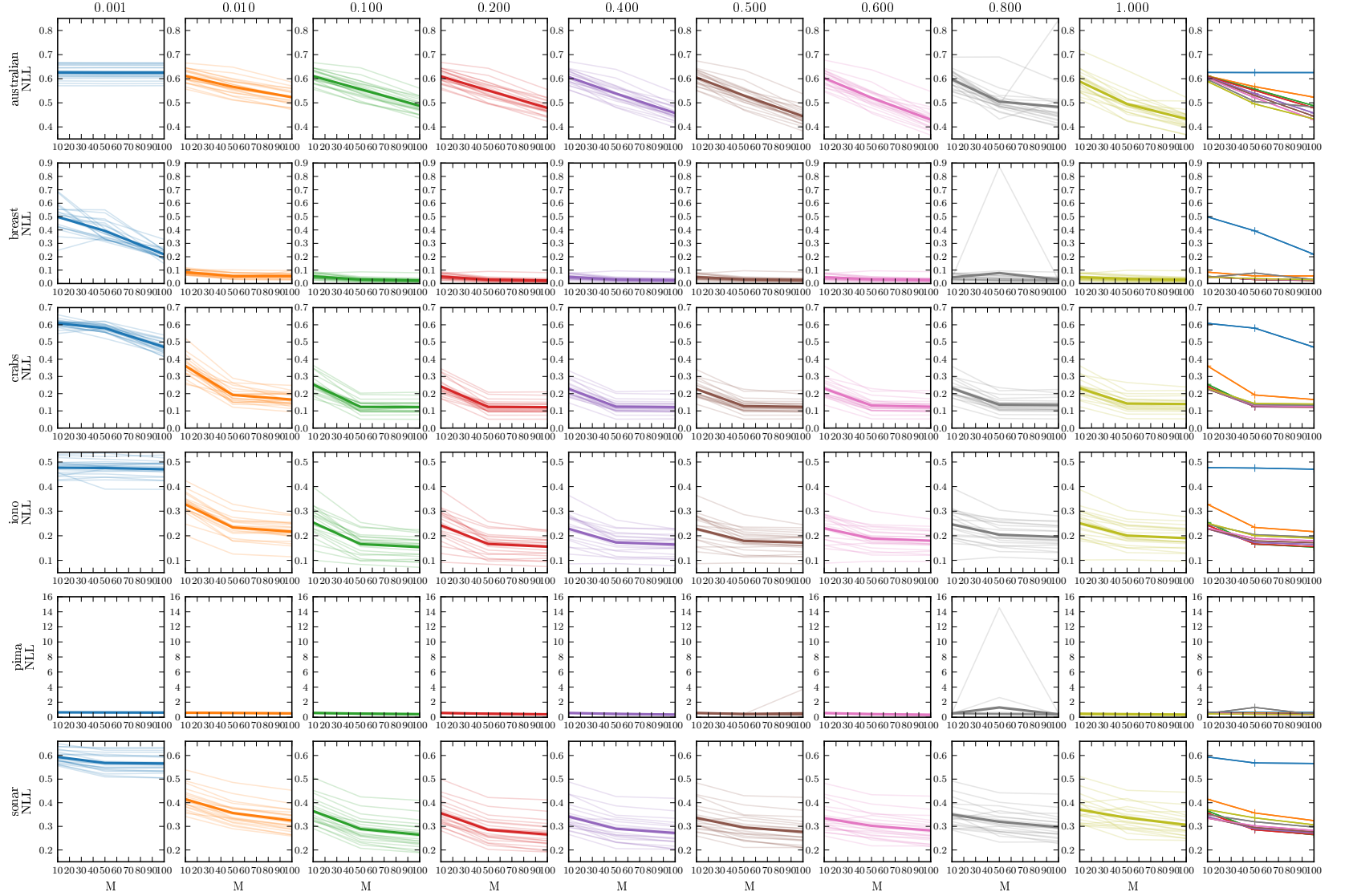


Figure 30: Results on real-world classification problems: Average negative log-likelihood on the test set for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ . Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various  $\alpha$  for comparison. Lower is better.

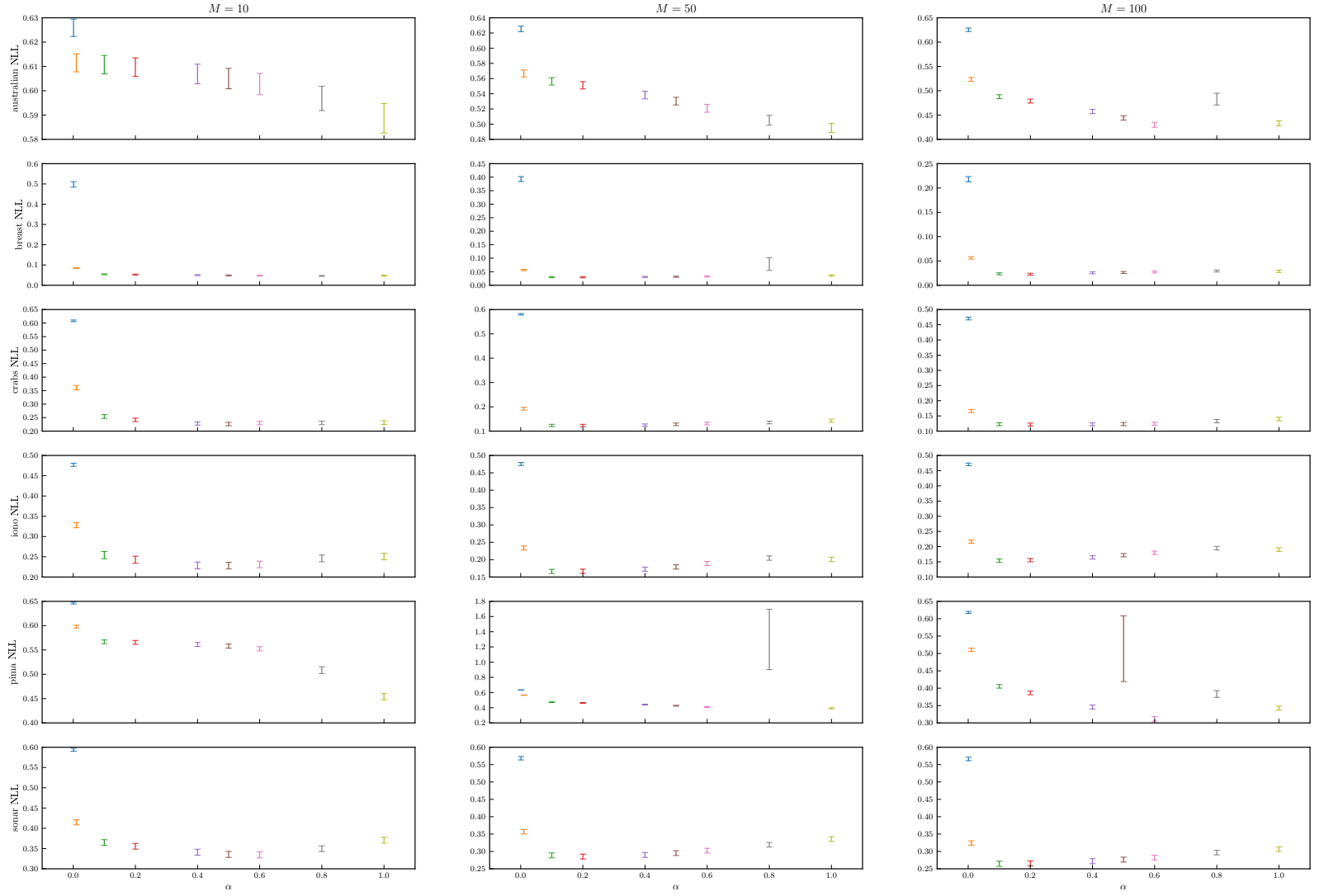


Figure 31: Results on real-world classification problems: Average negative log-likelihood on the test set for different datasets, various values of  $\alpha$  and various number of pseudo-points  $M$ , averaged over 20 splits. Lower is better.



#### G.4 Binary classification on even/odd MNIST digits

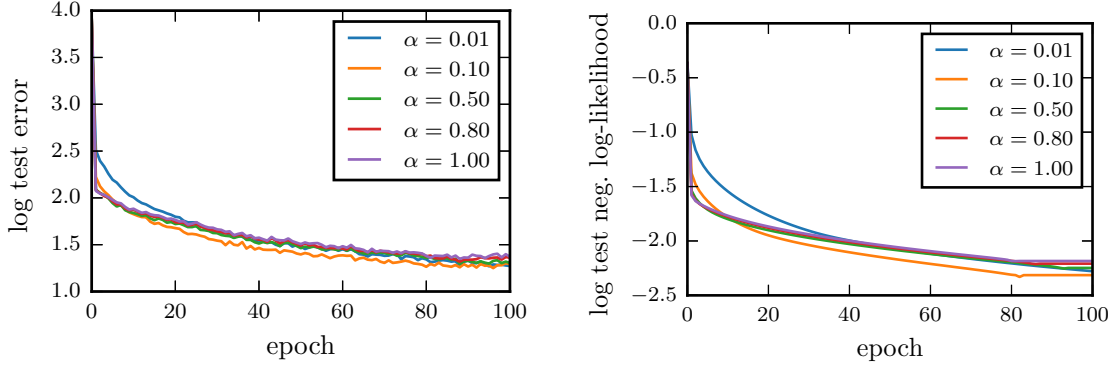


Figure 32: The test error and log-likelihood of the MNIST binary classification task (M=100).

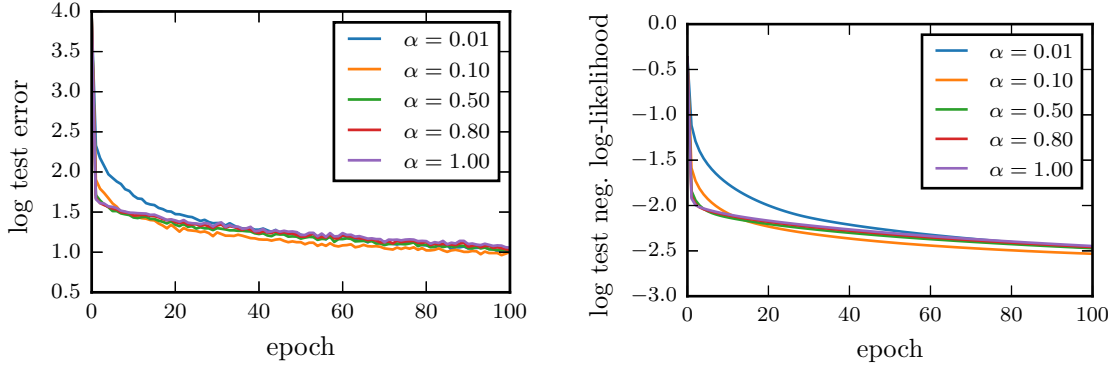


Figure 33: The test error and log-likelihood of the MNIST binary classification task (M=200).

#### References

- Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems 29*, pages 1525–1533, 2016.
- Thang D Bui and Richard E Turner. Tree-structured Gaussian process approximations. In *Advances in Neural Information Processing Systems 27*, pages 2213–2221, 2014.
- Thang D. Bui, José Miguel Hernández-Lobato, Yingzhen Li, Daniel Hernández-Lobato, and Richard E. Turner. Deep Gaussian process for regression using approximate expectation propagation. In *33rd International Conference on Machine Learning*, 2016.

- Lehel Csató. *Gaussian Processes — Iterative Sparse Approximations*. PhD thesis, Aston University, 2002.
- Lehel Csató and Manfred Opper. Sparse online Gaussian processes. *Neural Computation*, 14(3):641–669, 2002.
- Michael RW Dawson. *Understanding cognitive science*. Blackwell Publishing, 1998.
- Marc Deisenroth. *Efficient Reinforcement Learning using Gaussian Processes*. PhD thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2010.
- Marc Deisenroth and Shakir Mohamed. Expectation propagation in Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems 25*, pages 2609–2617, 2012.
- Amir Dezfouli and Edwin V Bonilla. Scalable inference for Gaussian process models with black-box likelihoods. In *Advances in Neural Information Processing Systems*, pages 1414–1422, 2015.
- Anibal Figueiras-Vidal and Miguel Lázaro-Gredilla. Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, pages 1087–1095, 2009.
- Roger Frigola, Yutian Chen, and Carl E. Rasmussen. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems 27*. 2014.
- Andrew Gelman, Aki Vehtari, Pasi Jylänki, Christian Robert, Nicolas Chopin, and John P Cunningham. Expectation propagation as a way of life. *arXiv preprint arXiv:1412.4869*, 2014.
- Agathe Girard, Carl E. Rasmussen, Joaquin Quiñonero-Candela, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs — application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems 15*, pages 529–536, 2003.
- James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, pages 282–290, 2013.
- James Hensman, Alexander G. D. G. Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *18th International Conference on Artificial Intelligence and Statistics*, May 2015.
- Daniel Hernández-Lobato and José Miguel Hernández-Lobato. Scalable Gaussian process classification via expectation propagation. In *19th International Conference on Artificial Intelligence and Statistics*, 2016.
- Trong Nghia Hoang, Quang Minh Hoang, and Kian Hsiang Low. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *32nd International Conference on Machine Learning*, pages 569–578, 2015.

- Trong Nghia Hoang, Quang Minh Hoang, and Bryan Kian Hsiang Low. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *33rd International Conference on Machine Learning*, 2016.
- Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.
- Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.
- Neil D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.
- Kian Hsiang Low, Jiangbo Yu, Jie Chen, and Patrick Jaillet. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *29th AAAI Conference on Artificial Intelligence*, pages 2821–2827, 2015.
- M. Mahsereci and P. Hennig. Probabilistic line searches for stochastic optimization. In *Advances in Neural Information Processing Systems 28*, pages 181–189, 2015.
- Alexander G. D. G. Matthews, James Hensman, Richard E Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *AISTATS 19*, 2016.
- Andrew McHutchon. *Nonlinear modelling and control using Gaussian processes*. PhD thesis, University of Cambridge, Department of Engineering, Cambridge, UK, 2014.
- Thomas Minka. Power EP. Technical report, Microsoft Research, Cambridge, 2004.
- Thomas P Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.
- Andrew Naish-Guzman and Sean B. Holden. The generalized FITC approximation. In *NIPS*, pages 1057–1064. Curran Associates, Inc., 2007.
- Hannes Nickisch and Carl E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078, 2008.
- Yuan Qi, Ahmed H. Abdel-Gawad, and Thomas P. Minka. Sparse-posterior Gaussian processes for general likelihoods. In *26th Conference on Uncertainty in Artificial Intelligence*, 2010.
- Joaquin Quiñonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.
- Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

- Jaakko Riihimäki, Pasi Jylänki, and Aki Vehtari. Nested expectation propagation for Gaussian process classification with a multinomial probit likelihood. *Journal of Machine Learning Research*, 14(Jan):75–109, 2013.
- Anton Schwaighofer and Volker Tresp. Transductive and inductive methods for approximate Gaussian process regression. In *Advances in Neural Information Processing Systems*, pages 953–960, 2002.
- Matthias Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9(Apr):759–813, 2008.
- Matthias Seeger and Michael Jordan. Sparse Gaussian process classification with multiple classes. Technical report, Department of Statistics, University of Berkeley, CA, 2004.
- Matthias Seeger, Christopher Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Artificial Intelligence and Statistics Conference 9*, 2003.
- Edward Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, University College London, 2007.
- Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 19*, pages 1257–1264, 2006.
- Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959, 2012.
- Michalis K. Titsias. Variational model selection for sparse Gaussian process regression. Technical report, School of Computer Science, University of Manchester, 2009a.
- Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *12th International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009b.
- Michalis K. Titsias and Neil D. Lawrence. Bayesian Gaussian process latent variable model. In *13th International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.
- Felipe Tobar, Thang D. Bui, and Richard E. Turner. Learning stationary time series using Gaussian processes with nonparametric kernels. In *Advances in Neural Information Processing Systems 29*, 2015.
- Richard E. Turner and Maneesh Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press, 2011.
- Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems 18*, pages 1441–1448, 2005.

Minjie Xu, Balaji Lakshminarayanan, Yee Whye Teh, Jun Zhu, and Bo Zhang. Distributed Bayesian posterior sampling via moment sharing. In *Advances in Neural Information Processing Systems 27*, 2014.

Huaiyu Zhu and Richard Rohwer. Measurements of generalisation based on information geometry. In *Mathematics of Neural Networks*, pages 394–398. Springer, 1997.